5-2015

Leveraging the BLAC Model for Situation Awareness in CIP

Samah Mobarak Balkhair

Leveraging the BLAC Model for Situation Awareness in CIP

by

Samah Mobarak Balkhair

A Thesis Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. Rajendra K Raj

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

May 2015

The thesis "Leveraging the BLAC Model for Situation Awareness in CIP" by Samah Mobarak Balkhair has been examined and approved by the following Examination Committee:

Dr. Rajendra K Raj
Professor
Thesis Committee Chair

Dr. Carol Romanowski
Associate Professor

Dr. Jennifer Schneider
Professor

Dedication

TO THE GREATEST MAN IN MY WORLD,
MY FATHER, MOBARAK BALKHAIR,
WHO WAS GONE BEFORE I FINISHED.

Acknowledgments

First and foremost, I am grateful for God for all His countless blessing upon my life, including having the opportunity to pursue and finish this research.

Surely, I am thankful for my advisor, *Dr. Rajendra K Raj* , and all committee members, *Dr. Carol Romanowski* and *Dr. Jennifer Schneider*, for their help, support and encouragement. I also don't want to forget to mention *Dr. Hans-Peter Bischof*, the Graduate Program Director, for his advising during my journey in master program. Also, I want to thank *Cindy Wolfer*, the Academic Advisor, for her lovely attitudes to relieve the stress.

My utmost gratitude for my parents, *Mobarak Balkhair* and *Najia Noah*, for all their love, support, and prayers. My siblings, *Wail*, *Rania*, *Loay*, *Sohaib*, and *Afraa* were, and always are, there to help and support me.

Last, but definitely not least, no words can express my sincere gratefulness for my husband, *Rami Batteh*, and my children, *Shaimaa* and *Omar*, for all their support, sacrifices, love, and understanding.

Abstract

**Leveraging the BLAC Model
for Situation Awareness in CIP**

Samah Mobarak Balkhair

Supervising Professor: Dr. Rajendra K Raj

Because critical infrastructure (CI) is considered to be the backbone of a nations livelihood, the Critical Infrastructure Protection (CIP) program was established in 1998 by the American President as a directive to ensure the United States security and to keep it safe from any malicious attacks or unexpected disasters. However, protecting the critical infrastructure of this country has always been challenging, especially with the interdependencies among each sector, which increases its vulnerability. Any failure of one sector could cause a significant impact on the others. The Decision Support System (DSS) is a computer system that helps agencies make decisions based on existing situations. The decision making of the DSS has a strong relationship to situational awareness (SA), such that the quality of a decision relies on the degree of the situational awareness. Regarding CI, any decision for one sector affects other sectors. Therefore, decision makers should have insight about the consequences for all sectors when one sector changes.

To avoid the complexity of interdependencies in the critical infrastructure of the United States, the healthcare sector was chosen to demonstrate the thesis hypothesis. Situational awareness is applied to healthcare, in general, but this thesis focuses on the situational awareness that is addressed during a multi-casualty incidents (MCI).

Bi-layer access control (BLAC) is an instrument that was proposed recently in a doctoral dissertation. BLAC uses a new concept named pseudoroles to verify the eligibility of access. For any request, BLAC checks, as a first step, against fixed attributes, then, in the next step, it matches against the associated policies. This two-step process increases the efficiency of the method. Although BLAC was introduced into the healthcare sector to protect patients privacy by regulating access to electronic health records, the pseudoroles concept can be implemented in any other sector.

This thesis aims to utilize the concept of pseudoroles leveraged in situational awareness in the CIP program. Focusing on MCIs, this thesis proposes a model that will help ambulance personnel make decisions as to the most appropriate hospital for each patient's care. The decision making depends on multiple factors that are related to the patient's case and the policies of each hospital. Because of the importance of time, in such situations, and the need for the patient to be transported to a hospital as soon as possible, the model uses the BLAC mechanism of a two-step evaluation to exclude all impractical hospitals, at a first glance, based on the hospitals' policies. Thus, by complex evaluation of the hospitals' information, there is a higher probability that the most appropriate hospital will be chosen for the patients.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Critical infrastructure (CI) is considered the foundation of a nation's livelihood. A nation's health, wealth, and security rely on certain goods and services that are produced by these critical infrastructures [19]. As a result, protecting critical infrastructure from natural disasters or any malicious attack is considered of utmost importance. However, the problem with CIs is their interdependency, which makes each one depend upon all of the others [5]. Any damages or lessening of services in any sector can affect the functioning of the other sectors. In 1996, the United States established the Committee on Critical Infrastructure Protection (CCIP) [5]. In 1998, the Presidential Decision Directive was released, and it established the Critical Infrastructure Protection (CIP) program [19]. Since then, researchers keep working on the safety and security of this countrys CI.

Situational awareness (SA) is an old concept that was used in the military. It has received more renewed over the past few decades to extend the concept to other fields [24]. Wide-area situational awareness (WASA) is a new concept that was derived from the SA concept for particularly critical infrastructure protection. The National Institute of Standards and Technology in the US classifies WASA as one of the eight priority areas to protect CI [2]. WASA was proposed as a general framework for CI protection, but there are other systems that specific CI protection, such as smart grids and water supply network [1].

One of the SA-related models is decision making, which is needed for all different CIs

at multiple levels. The decision makers should make their decisions based on knowledge about all of the options available to them and all of the consequences that will result from their decisions. In a complex and dynamic environment, decision making relies on SA because the environment can be affected if SA decisions or outcomes are not accurate [11]. The Decision Support System (DSS) is a computer-based system that helps in decision making by taking the best available decision and providing all options based on collected information from different sources. There is a special DSS that is designed partially for CIs, named CIPDSS [7]. CIPDSS helps decision makers take the most risk-mitigating decision by considering all other infrastructures. Detecting the affected infrastructures from any critical event that impacts certain infrastructure is challenging, especially when time plays a role in mitigating the risk.

To avoid the complexity of interdependencies in the critical infrastructure of the United States, the healthcare sector was chosen to demonstrate the thesis hypothesis. Situational awareness is applied to healthcare, in general, but this thesis focuses on the situational awareness that is addressed during a multi-casualty incident (MCI). The thesis proposes a model that will help ambulance personnel make decisions as to the most appropriate hospital for each patients care. The decision making depends on multiple factors that are related to the patient's case and the policies of each hospital. Because of the importance of time, in such situations, and the need for the patient to be transported to a hospital as soon as possible, the model uses the BLAC mechanism of a two-step evaluation to exclude all impractical hospitals, at a first glance, based on the hospitals policies. Thus, by complex evaluation of the hospitals' information, there is a higher probability that the most appropriate hospital will be chosen for the patients.

In this thesis, an empirical data was used with fabricated policies to elucidate the proposed model. Regardless to the data volume and the realistic of the policies, the proposed model can be applied in any CI sectors. The essential point is to declare the most important elements in and define the policy that should be taken during the emergency cases. Most importantly, the criteria that plays a role in the emergency should be defined to make it

changeable during the crisis time.

1.2 Background

1.2.1 Critical Infrastructure Protection (CIP)

Modern societies rely on a set of continuous and reliably available services [16], such as the electricity, water supply, transportation, and communication systems. Any disruption or damage infecting one or more of these services for a significant period of time would have a negative impact on our nation's livelihood. These services, which are the backbone of our nation's life, are called critical infrastructures [16]. The USA Patriot Act of 2001 defined critical infrastructures as "systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters" [5] The Department of Homeland Security identified 16 sectors for consideration:

- chemical facilities
- commercial facilities
- communication
- critical manufacturing
- dams
- the defense industrial base
- emergency services
- energy
- financial services

- food and agriculture
- governmental facilities
- healthcare and public health
- information technology
- nuclear reactors
- materials and waste
- transportation systems
- water and wastewater systems .

These sectors are not only critical in their own right, but they are also critical in their interdependencies [16]. These sectors have bidirectional relationships between each other and the status of one infrastructure can be affected by, or correlated with, the state of another [5]. The interdependencies are classified as either physical, cyber, geographic, or logical [18]

1.2.2 Situational Awareness (SA)

Situational awareness is a human mental process that makes people in charge aware of the various situations that could happen in their environment by knowing and understanding what is happening around them. Situational awareness is acquired by using a diversity of cognitive processes and searching about and accessing all possible information about users, data, systems, and environmental sensors in order to have a comprehensive perspective about the mission. To achieve effective situational awareness, a myriad of technologies should be integrated to provide and analyze comprehensive information about specific situations, which are then sorted and prioritized based on the mission requirements and then produced as an output in the user interface.

Situational awareness is a flexible concept that can be used differently in each sector. Originally, the concept of situational awareness was used in military pilot training[11], but it is important in many fields, even though it could be used under other names. Therefore, there are various definitions of the term situational awareness. For example, SA can be defined as "a state of knowledge, from the processes used to achieve that state" or "a variety of cognitive processing activities" [10]. Also it can be defined as "being aware about what happens around you and understanding what info means to you now and in the future" [11]. Therefore, some researchers think that developing a definition for situational awareness is useless [10]. However, Endsley [10] stressed that it is necessary to clearly define SA to produce significant progress. Based on the previous definitions, all of them agree that situational awareness is "knowing what is going on" [10]. Hence, Endsley [10] proposed a formal definition for SA, which is "SA is a perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and projection of their status in the near future."

The definition of SA, as Endsley [10] defined, is broken down into three levels. Figure 1.1 shows Endsleys model:

- Level 1: **Perception**

  The perception level is simply an answer to the question, "What is the current situation?" [25]. Without having this information, the anomalous events will not be detected [25]. The requirements of this level are quite different, depending on the domain in which SA is applied. The goal of this level is to perceive the statues, attributes, and dynamics that are relevant elements in the environment [10][11]. Information perception is gained through one or more of the human senses, such as sight, sound, touch, taste or smell [11]. Since it requires all of the needed information to consider a difficult task, most SA errors occur at this level [11].

- Level 2: **Comprehension**

  The comprehension level answers the question, "What is actually going on?" [25].

This question can be answered by understanding the relationship between the different data, which is acquired in the previous level, depending on the relevant goal. This level involves integrating, sorting, and prioritizing the pieces of the data to extract the meaning. Some error could occur at this level when the practitioner has the necessary data but is not able to understand the meaning of the information provided. [11].

- Level 3: **Projection**

  The projection level answers the question, "What is most likely to happen if . . . ?"[25]. At this level, the practitioner should be able to predict the elements that are defined in Level 1 and understand their meaning in relation to Level 2, and decide what will need to be done in the future [11]. If the two previous levels are processed accurately, there is a small probability that the projection level will fail to provide accurate projection information[11].

Endsley's model of SA was extended to include **Resolution** a level that aims to identify the approach for the current situation [25]. The resolution level helps the practitioner answer the question, "What exactly shall we do?" by providing the end of all available options and how it affects the environment [25].

Decision Support System (DSS):

After practitioners analyze, manipulate, and evaluate various complex factors, the DSS is an interactive computer-based system that helps them make decisions based on the current situation [15]. Decision making has a strong relationship with SA because the quality of a decision relies on the degree of the SA [24]. An empirical study [24] briefed a number of models that were designed for this purpose. Rather than giving a certain decision, an Unbounded Rationality model provided all possible options with their consequences, leaving the action to the decision maker. Although this model was convincing, theoretically, it was not practical in the real world where the decision makers were limited on time, knowledge, and computational capacity. On the other hand, the Optimization Under Constraints model was similar to the previous model, but it had stop rules that would end its research when it

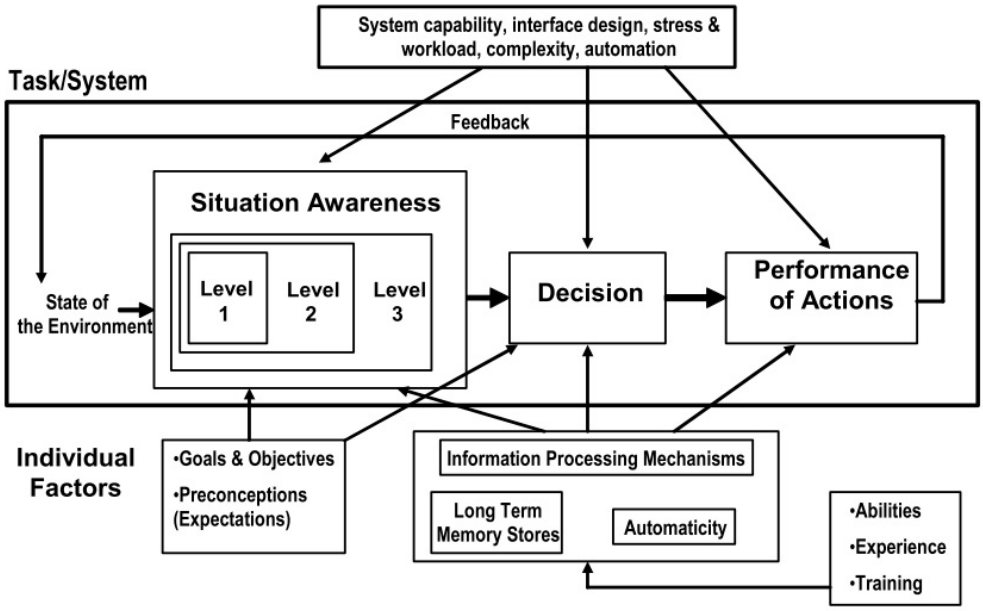

Figure 1.1: Model of SA in dynamic decision making [11]

reaches certain points. Finally, Naturalistic Decision making aims to reduce human interaction by analyzing the daily decisions. This model compares the situation with previous decisions that were made and follows the decision if it matches one of them.

1.2.3 Bi-Layer Access Control (BLAC)

The Bi-Layer Access Control (BLAC) is an access control model that was recently proposed in a PhD dissertation [3]. BLAC is designed mainly to regulate access to Electronic Health Records (EHR) in the healthcare sector and to achieve the Health Insurance Portability and Accountability Act (HIPAA) requirements. BLAC was developed in such a way that it combines the advantages of two previous models: Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC). While RBAC provides a simple access control management, it involves a complex initial role and makes access control inflexible [3].

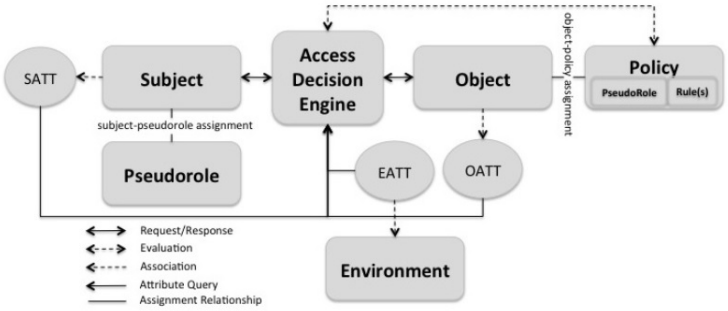

Figure 1.2: The BLAC Model

On the other hand, ABAC overcomes inflexibility and simplifies the initial setup. However, the complexity of managing the privileges and user permissions is increased. The BLAC model was designed to support attributes and policies. In BLAC, subjects are associated with pseudoroles, which are composed of a set of static attributes, and objects are associated with policies, which specify how attributes are to be considered for access requests [3]. BLAC provides a two-step process to check whether the subject is eligible to access a certain object. First, it checks whether the subject has the required pseudorole or not; if it does, it checks the rules within the policies to give a final decision to grant or deny access [3]. This two-step process increases the efficiency of the method when it denies access from the first step without any further checking. Although it was proposed in the healthcare sector, BLAC could be used and implemented in any mechanism to improve its efficiency by using the concept of the pseudorole.

1.3 Related Work

1.3.1 Situational Awareness Model

As pointed out previously, SA is a broad concept that could be formulated based on a target field. Different models can be developed for the SA system [1]. One of these model is the Joint Directors of Laboratories (JDL) model, which is a layered hierarchical model [1].

The JDL model has levels that match the levels of Endsleys SA model. JDL has a flexibility that enables it to deal effectively with data that comes from different CIs. However, it just highlights faults and failures to the operators by providing a snapshot of the system [1]. The Boyd Control Loop (BCL) is another model of SA [1]. This model consists of four phases: Observe, Orient, Decide, and Act, which explains the reason that it is called the OODA loop. In Observe phase, the information gets collected from the environment, statuses, and threats, and the Orient phase makes estimates, assumptions, analysis, and judgments on the situation. After that, the Act phase determines the needed action to pass it to the Decide phase that turns decisions into Actions. This model allows the operator intervene in the Act phase to end the loop by transferring the decision into real action. The Observe, Orient, and Decide phases correspond to Endsleys SA model.

Critical infrastructure has a critical and distributed nature that needs a specific model designed for it. Critical infrastructures cover a wide range of sectors [5]; therefore, each sector need a specific SA system that is designed partially for its requirements [1]. However, because of their interdependent and distributed nature, CIs share some common requirements [1]. For example, situational awareness in CIP allows control systems, which are responsible for monitoring critical infrastructure, to detect any malicious incidents and threats occurring within a critical infrastructure at any given time [2]. In the CI field, human supervision is required to ensure immediate and efficient responses for any internal failures or intentional attacks. However, human supervision is not always guaranteed, especially for these infrastructures, which are located where human control might be minimal or even nonexistent [2]. To overcome this issue, the National Institute of Standards and Technology (NIST) derived a new concept of situational awareness named it Wide-Area Situational Awareness (WASA) [2]. WASA is a model of SA that aims to ensure dynamic prevention and response services by increasing the effectiveness of technology to make human monitoring possible anywhere and anytime. The security system for WASA is concerned not only with confidentiality, but also with availability and integrity because the absence of these two properties, resource unavailability or variations in their content, could trigger a

major security risk. Given that WASA utilizes various technologies, it could produce security threats to availability, integrity, and confidentiality (AIC) if it was not implemented properly. Moreover, it could introduce incompatibilities, conflicts, or operational delays into these technologies that work together regardless of their location and environmental conditions. All of these threats affect negatively on a system's survival.

The proposed WASA framework integrates two theoretical concepts: the hybrid perspective, which requires a human presence to monitor emergency situations, and the context-awareness concept for the protection of functional services [2], which is considered as a basis of both proactive and reactive solutions [1]. WASA contains two main phases: the setup and commissioning phase, and the development phase. In the first phase, the context is specified by defining the primary structure, which includes four attributes: location, identity, activity, and time [2]. The second phase is the actual set of processes during life cycle which consist of six actions [2]:

1. normalization
2. prevention and detection
3. location, alerting, and display
4. response and recovery
5. learning and updating
6. assessment and reporting.

1.3.2 Critical Infrastructure Protection Decision Support System (CIPDSS)

Decision making within the CI's issues is crucial and complex because it involves sector interdependencies. The Department of Homeland Security and Technology Directorate fund and the Critical Infrastructure Protection Decision Support System (CIPDSS) [7] are studying this important issue [12]. CIPDSS helps decision makers make decisions by informing

them about any possible risk, considering all critical infrastructures and their interdependencies [12] [22]. CIPDSS is a combination of software programs, analysis procedures, and decision support tools that are designed to simulate scenarios and estimate their consequences [22]. It uses a case-study framework that contains at least two scenarios: a base scenario and one or more alternative scenarios [23]. Each scenario is associated with a readiness scenario and an incident scenario.

Initiated as a proof-of-concept prototype in August 2003, the CIPDSS project completed a model and two case studies in February 2004 [7]. The main goal of CIPDSS was to provide insight for making decisions that were related to critical infrastructure protection by considering all other critical infrastructures [7]. The architecture of CIPSDSS includes consequences models of all critical infrastructures [7] (see Figur 1.3).

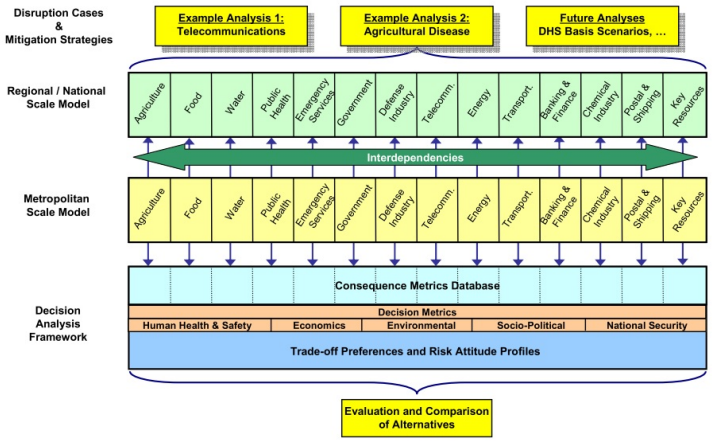

Figure 1.3: CIPDSS architecture [7]

1.3.3 Simulate Critical Infrastructure Interdependencies

Although there are several simulation models for individual infrastructures, these models do not capture the emergent behavior that arises from interdependency [18]. Moreover, each model has its own requirements and functionality. Therefore, Hyeung-Sik J. *et al* [18] worked on integrating existing individual models to produce a model that dealt with

different models' requirements. Their models have been built for a number of individual infrastructures including power, petroleum, natural gas, water, transportation, and communications.

1.4 Hypothesis

This thesis aims to demonstrate the ability of the Bi-Layer Access Control (BLAC) concept to be used in critical infrastructure to enhance the approach of situational awareness. The main concept of BLAC is to assign subjects to static attributes and assign objects with policies. Then the access control is performed in a two-step evaluation procedure [3]. The BLAC mechanism provides flexibility to change the policy of the object in order to change the target that could access a particular object. In the same way, the goal of the thesis is to utilize the concept of assigning static attributes and policies to objects that are parts within a critical infrastructure to provide flexibility in situational awareness in that infrastructure. The hypothesis of the thesis is the BLAC model provides a flexible approach to handling situational awareness in CIs and increases the possibility of making a correct decision by using a portion of the total information. It attempts to provide a model that utilizes the BLAC concepts (a two-step evaluation procedure) in CI's models. However, doing the analysis of such a model is challenging because of several factors [18]. First, obtaining the data is difficult. Second, each individual infrastructure is complicated. Moreover, infrastructures are constantly evolving and governing regulations are constantly changing. Nevertheless, the hypothesis of this thesis can be verified by considering a single infrastructure, which will be demonstrated in next section 1.5.

1.5 Motivation

Focusing in on the healthcare sector, the Emergency Department (ED), also known as Emergency Room (ER), is usually found in hospitals and health care centers. It is mainly responsible for unplanned crucial medical cases; therefore, the ED staff should be ready

to provide any initial treatment needed for any case. Moreover, ED staff can observe life-threatening cases that need specialist physicians and certain resources. Crowding in the ED is considered a significant problem because it affects the efficiency of staff functioning. Crowding in the ED is defined in [14] as *when the identified need for emergency services exceeds available resources for patient care in the emergency department, hospital, or both"*. Generally, ED crowding is the result of a multiple-casualty incident, when more than one person is injured because of one incident, such as natural disaster, terrorist bombing, or bus crash. To avoid crowding in any one ED and to provide the best service to the victims, patients should be distributed effectively amongst the hospitals EDs that are located in the incident area, and they should be kept aware of the changing statuses of what is happening in each ED.

The most important step that should be taken when a multiple-casualty incident has taken place is to triage the people who are affected and who are in need of help [6]. The term triage is used for the decision-making process and for distributing medical resources among patients [9]. It is a process of categorizing patients, depending on the severity of their medical condition, and prioritizing the treatment among them. The goal of triage is to use the limited medical resources widely to save as many lives as possible. Therefore, a triage decision should be made in a short amount of time that ensures each patient receives his/her need of treatment in the right amount of time [21]. However, ED overcrowding is considered a threat to patient safety and public health. Accordingly, this thesis aims to propose a model that helps to mitigate overcrowding in EDs by using BLAC characteristics.

In terms of BLAC characteristics, the patients and the hospitals are the important elements in MCIs. Therefore, the assignment attributes and policies should be for MCIs. Likewise the BLAC model makes a decision to grant or deny a subject (user) access to an object (document). The proposed model would consider the hospital and patient as a subject and object to make a decision whether the patient was eligible to access a certain hospital. On this basis, the patients would be assigned to attributes and the hospitals would be assigned to policies.

1.6 Roadmap

This thesis is organized in the following manner:

- **Chapter 2:** The thesis proposes the model for MCIs. In this chapter, the details are explained for the complete process of the model to distribute patients among hospitals by assuming a certain scenario and showing the procedure that the model takes to reach to certain decision for each patient.
- **Chapter 3:** A part of the model has been implemented. This chapter contains the details of implementation. It shows the three layers of the model: web browser, web server, and database server.
- **Chapter 4:** In this chapter, the details of the experiment are elucidated, and the analysis of the results is discussed.
- **Chapter 5:** Conclusions from the research are discussed and anticipated future work is outlined.

Chapter 2

Design

In this chapter, the proposed model will be explained in a specific scenario. The scenario that is assumed here is when an incident happens that causes multiple-casualties in a certain area. With a large number of patients severely ill or injured, one of the triage methods should be used. Triage is a term in the medical field that is defined as the process of categorization of casualties based on their need for medical attention" [17].

There are different triage techniques that are used around the world. One of the triage methods that is widely used in North America is known as START (Simple Triage and Rapid Treatment) [13]. In START, victims are classified into four categories, depending on the urgency of their need for evacuation. The categories are: green for a delayed care; yellow for urgent care; red for immediate care; and black for deceased or those who are not expected to survive due to their extensive injuries [9]. Indeed, triage decision making is a very complex task because it is based on uncertainties and ambiguous information [21]. Nevertheless, this thesis focuses on the decision making after the patient has been classified by a rescue worker. The thesis proposes a model that will help ambulance personnel determine the most appropriate hospital for transporting a particular patient. The goal of the model is to distribute patients effectively among the nearby hospitals to mitigate the overcrowded situation in any ED.

Based on the assumed scenario, a certain incident occurred, for example, a bus accident, in a specific location. As a result of this accident, multiple people have been affected and their injuries vary between very serious to minor. In the following section, the application flow will be explained, starting from the data that should be obtained and maintained before

any crisis, through the crisis time based on the scenario that was assumed.

2.1 Application Flow

Decision making relies on the quality of the data that is provided to the application. In the proposed model, there are two stages of data collection: prior to the crisis and in the time of the crisis.

2.1.1 Prior to the Crisis

One element that affects the decision of which hospital a patient should be transported to is the hospitals information, such as hospital location and bed capacity in its ED. This kind of information should be saved in the application's database and updated regularly. Additionally, each hospital should be associated with an inner node in the policy file. The policy should be contained in an Extensible Markup Language (XML) file. All hospitals are linked to one policy file but they each have a different node. A structure of the policy XML file is shown in Figure 2.1. The XML file consists of a root node called *BasicAttributes*, and a set of one or more inner nodes called *Hospital*. Each *Hospital* node represents a hospital that is saved in the database. The hospital id and name are specified in the node's attributes. Each *Hospital* node has three inner nodes: *tag*, *age*, and *diag*. The *tag* node defines the type of patients that could be observed in that particular hospital. As a result, a *Hospital* node could have more than one *tag* node. Similarly, the *diag* node represents the initial diagnosis for the patient and a *Hospital* node could have more than one *diag* node. Contrarily, the *age* node defines the minimum age of the patient that could be accepted at a particular hospital, so there should be only one *Age* node in each *Hospital* node. The *age* node could be defined as "Any" for unspecified ages. The policy file is used as a first-step evaluation. It simply verifies whether a patient has the right attributes, specified by hospital policy, to be accepted for care. If attributes match, the process goes to a more complex evaluation, which brings into consideration the instantaneous variables that are happing within the hospital at that given point in time. The advantage of two-step evaluation is that

it speeds the process by limiting a number of hospitals that are exposed to the complex evaluation. Moreover, it makes changing the policy during the crisis time quick and easy. The hospitals could change their policy based on their current situation, which is a step that indicates improved effectiveness of the system by skipping the hospital when it becomes overcrowded. To change any of the policies, the values of the hospitals nodes in the policy XML file would be changed.

```
<!-- The root node of the policy file -->
<BasicAttributes>

   <!-- Each Hospital's node represents a hospial in DB -->
   <Hospital id="1" name="Hospital Name">

        <tag>
           <!-- specifies the tag colors that the hospital would accept -->
           <!-- Each hospital node could have one or more tag node -->
        </tag>



        <age>
           <!-- specifies the minimal age that the hospital would accept -->
           <!-- Each hospital could have only one age node -->
           <!-- The value is either ANY or a number -->
        </age>


        <diag>
           <!-- specifies the initial diagnosis that the hospital would accept -->
           <!-- Each hospital could have one or more diag node  -->
           <!-- The value is either ANY, Cardiac, Seizures or Orthopaedic -->
        </diag>

   </Hospital>

</BasicAttributes>
```

Figure 2.1: The structure of XML Policy File.

2.1.2 In Crisis Time

Based on the scenario that is assumed, multiple people are affected in a bus accident and need an urgent treatment. Rescue workers arrive at the scene to do their job by helping victims and by prioritizing them depending upon their health condition. As mentioned previously, patients are categorized into four categories: green, yellow, red, and black based on the START triage method [9]. Traditionally, rescue workers use a standard physical

tag and place it around a patient's neck. This tag has the patient's information and the color code depending on the situation of the patient [17]. Afterward the initial tagging, ambulance personnel start evacuating the proper patients to the hospitals for treatment. Without a doubt, patients should get treatment as soon as possible. Although it plays a huge role in reducing the time of getting to treatment, road distance to the hospital is not the only factor. In some conditions, transporting a patient to the outermost hospital is the right decision because multiple factors are taken into consideration.

To demonstrate these multiple considerations, let us assume ambulance personnel start with patient 'X' who needs transport to a hospital as soon as possible. One of ambulance personnel inserts the patient's basic information into the application which is: the patient's name, date of birth, initial diagnosis, and color of tag category. Additional information is needed, such as the current location of the patient and the approximate maximum time (in minutes) that the patient could delay before getting a treatment. The information get processed and compares the patient with the hospitals information that was previously inserted into the database. The processing goal is to find the most appropriate hospital for patient 'X', depending on the patient's information, which was inserted, and the hospital's policy, which was stored previously. Consequently, the application provides the ambulance personnel with a decision as to where the patient should be transported, and the application puts the patient on a waiting list at the chosen hospital. From this point forward, the patients will be called by their triage colors.

2.2 Application Mechanism

The application process mimics the two-step evaluation in the BLAC mechanism [3], which provides velocity in decision making. The core idea of the two-step evaluation is that it excludes as many hospitals as possible that cannot help with the patients status, in the first step, and it retains the most appropriate hospitals that can help the patient in his or her current status. As Figure 2.2 shows, the application process begins to compare the patient's information with the nearest hospital, based on patient's current location. The first step

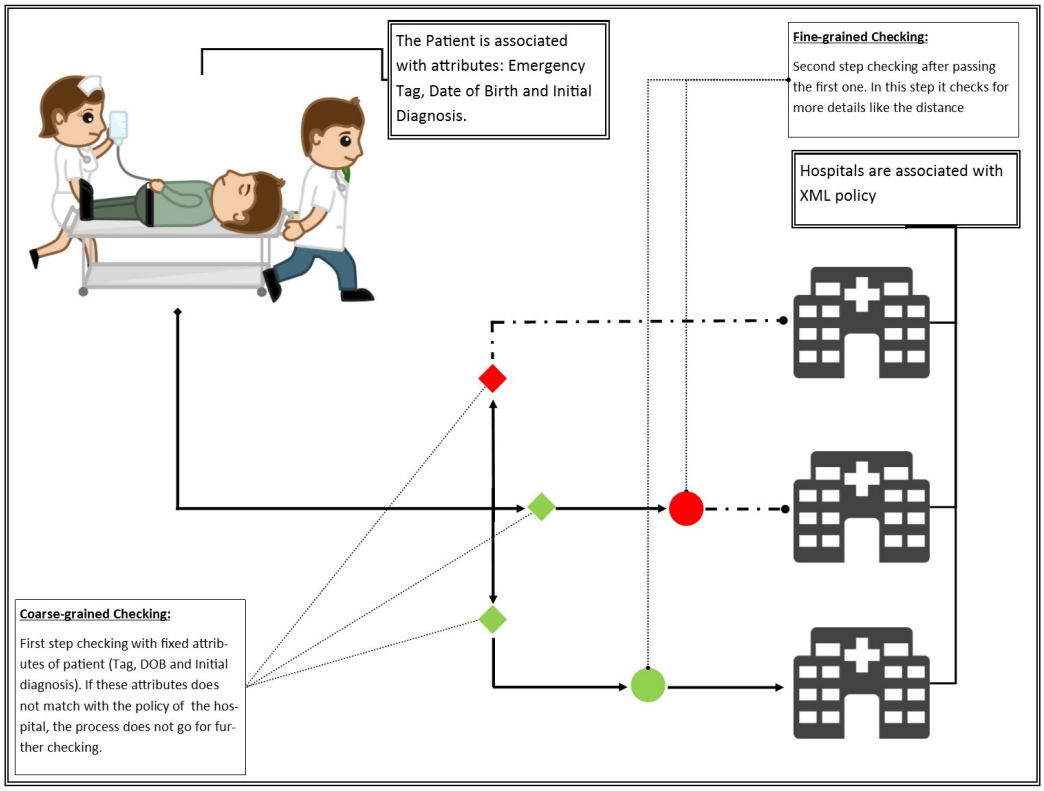

Figure 2.2: The perspective of the target model.

uses a simple comparison between the patient's basic attributes (initial diagnosis, date of birth, and triage tag color) and the hospital's policy. If these attributes do not match with hospital's policy, the hospital will be immediately excluded, and the application starts the comparison process with the second-nearest hospital. An example of excluding a hospital from a first step is when the patient is categorized as a red tag and the nearest hospital is not capable of handling life-threatening cases. In this case, the hospital indicates in its policy certain colors that it can observe. Therefore, the process will compare the patient's tag color with the variables of the hospital's policy in a *tag* node. Given that it did not find red, it excluded this hospital. Another example is when the patient is a child and the hospital does not have a pediatric section in its ED. Thus, the hospital's policy indicates that it only

accepts adult patients by specifying a minimum age. In this case, the policy includes a set minimum number of the *age* node. The same action goes with initial diagnosis when the hospital specifies a certain values in *diag* node.

On the other hand, if the basic information matches the basic attributes of the hospital's policy, the process goes further for another, more-complex, evaluation. While the first-step evaluation is concerned about the basic, fixed attributes, the second-step evaluation is concerned about the instantaneous variables about the hospital's status based on the incoming patients at that point in time. In this step, the system checks the road distance and the ratio between the number of patients who are currently on the waiting list and the hospital's bed capacity. After the application takes all this information into consideration, the system decides whether the hospital is appropriate for the patient to be transported. Figure 2.3 shows the basic process flow of the model.

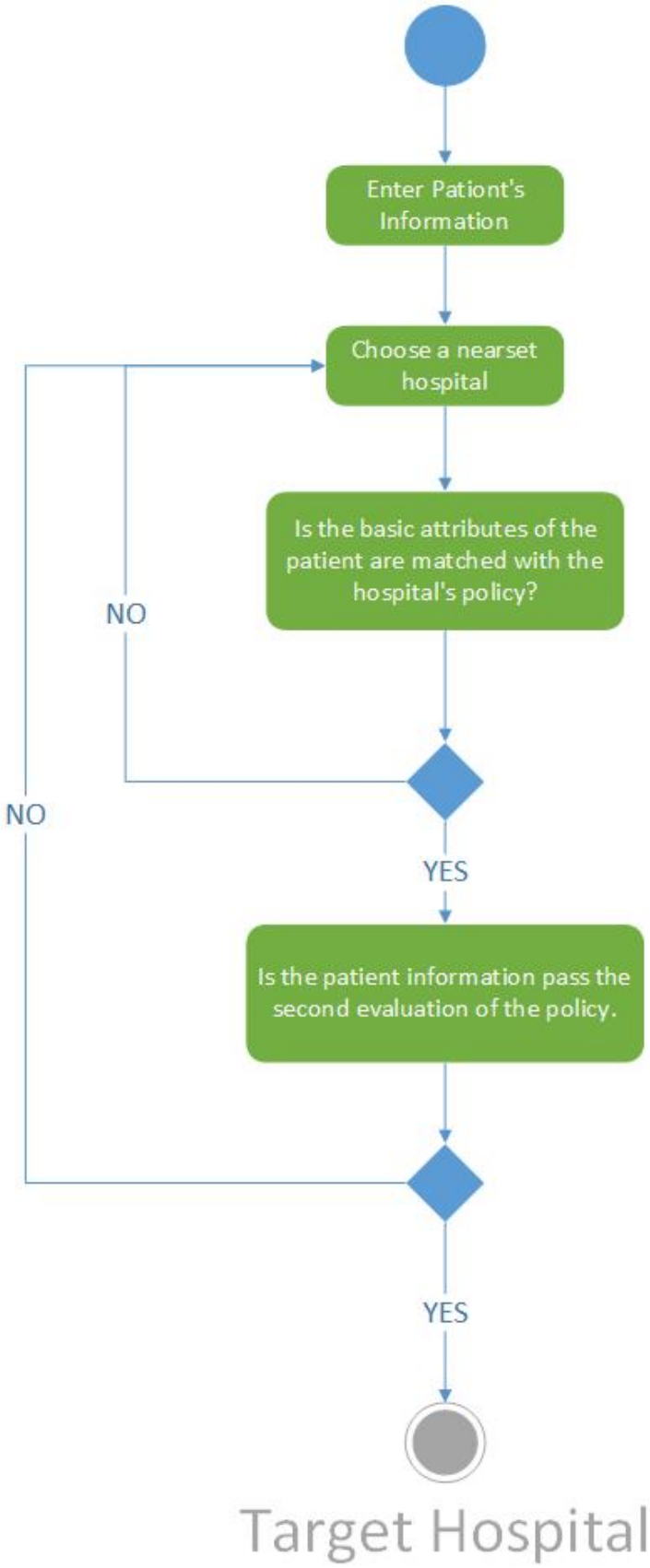

Figure 2.3: The application mechanism process.

Chapter 3

Implementation

A part of the proposed model has been implemented in order to verify the thesis's hypothesis. The implementation is a web application that is used in such an assumed scenario to find an appropriate hospital for the patients in a multiple-casualty incident. The web application consists of three layers: web browser, web server, and database server. In this chapter, the implementation of the web application is elucidated including all of its layers.

3.1 Web Browser

The users interact with the web browser to access the application. Therefore, it is so important to make it user friendly. The application interfaces were created using ASP.NET 2010. In the current implementation, only two interfaces are created that are relevant to the search process. The first interface is for obtaining hospital information. As explained in Chapter 2, this part of the process should be performed at an earlier stage and kept up to date. There are two kinds of information that are needed in this step of the process. First, the general information of the hospital is needed such as the hospital's name, address, bed capacity, and location. This information will be saved in the database server. The second part is related to the basic attributes of the hospital's policy. Each hospital should have defined the basic attributes for its policy by choosing values for each basic attribute: tag color, age, and initial diagnosis. This part of the information will be saved in XML file separated from the general hospital's information as it will be illustrated later on in this chapter. Figure 3.1 shows the hospital information interface.

The second interface is the main screen where the actual process is done. Moreover, it is the main screen that is used during the time of crisis. As shown in Figure 3.2, the patient information is entered into the first section, including the patient's name, tag color, initial diagnosis, date of birth, approximate maximum time that the patient could delay before getting treatment, and the latitude and longitude values of the patient's location. When the *Find Hospital* button is clicked, the application performs two processes. First, it saves the patient information in the database server, then it begins the search process for an appropriate hospital for this patient. It goes through some hospitals' information that are saved in the database after it excludes all hospitals that do not match the criteria of the patient's status. The exclusion process is done by using an XML policy file that has the hospitals' policies on it. The process goes through a list of the remaining hospitals that are sorted by the nearest, less crowded hospital. When the process finds an appropriate hospital, the information of the chosen hospital will appear in the second section. After any needed notes about the patient's case are written, the patient is included in the hospital's waiting list by clicking the *Reserve* button.

3.2 Web Server

The web server is mainly responsible for the business logic of the application. Visual Basic .Net is the language that is used in the implementation. In the hospital information interface, the application saves general hospital information in the hospitals table on the database server, followed by updating the XML policy file that is located on the web server. As shown in Figure 3.3, the policy file has a main root named *BasicAttributes*. Each hospital is represented by an inner node name *Hospital* and it has two attributes: id and name. When a new hospital is added in the application database, the application writes a new inner *Hospital* node in the policy file. The node holds the hospital's name and id as attributes. Moreover, the application defines three inner nodes for the new hospital node: *tag*, *age*, and *diag*. There could be more than one node for each one, depending on the values that are chosen in the interface. For example, there will be one *tag* node if the user

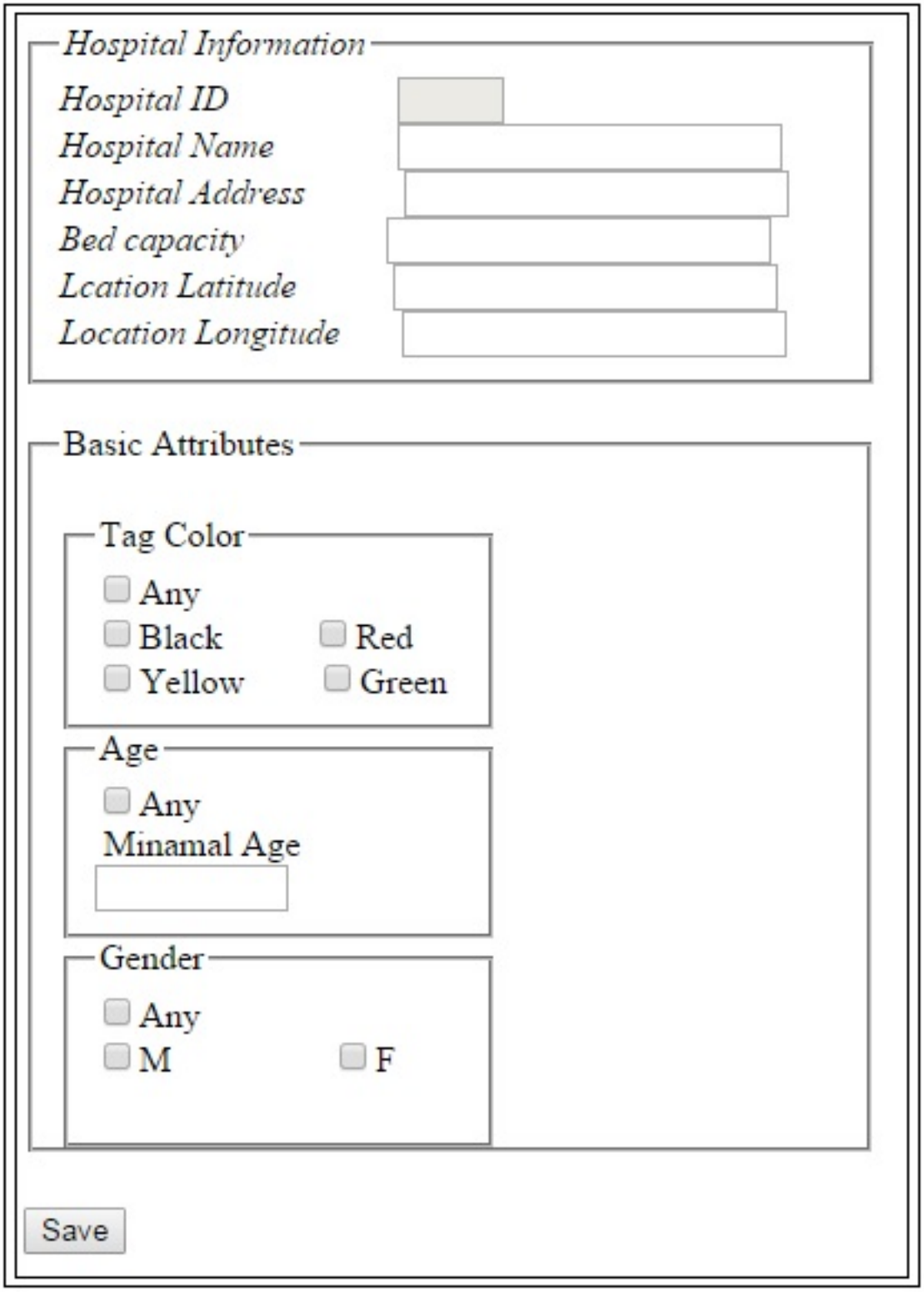

Figure 3.1: Hospital information interface.

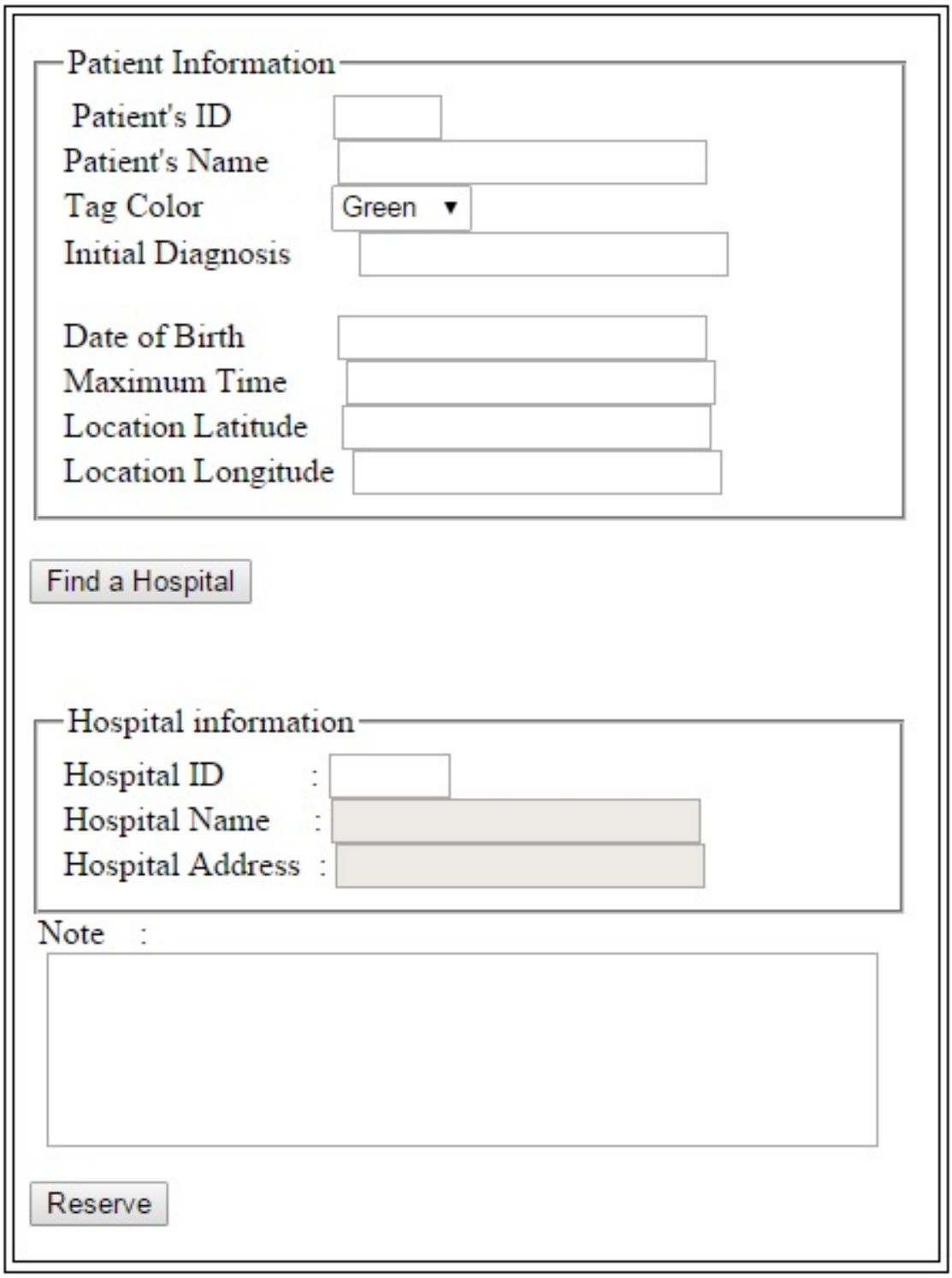

Figure 3.2: Patient information interface.

chooses ”Any” for the tag, and there will be two nodes named *tag* if both ”Red” and ”Yellow” are chosen. In other words, at least one inner node from each category is defined for each *Hospital* node.

The second part of the implementation is the search process. This process is done in the main interface where the patient's information is entered. The patient's information is saved on the database server in patients table. After saving the patient's information, the application accesses the XML policy file on the web server. The application uses the *XmlDocument* class in the *system.XML* namespace that represents an XML document [4]. The path of the XML file is specified in the *Load* method in order to make the data in XML file accessible. After accessing the data in the XML file, the application fetches all hospitals' IDs that match the basic attributes of the current patient. This step uses XML Path Language (XPath) [8], which is designed to have the ability to navigate in the XML tree and select nodes based on predefined criteria. XPath query is embedded in the *SelectNodes* method, which has the values of the nodes that are needed. Then, the values of the ID attributes are returned as a list in a string data type that is separated by commas. The ID list is used to fetch hospitals' information from the database server using SQL query.

For further clarification, an example of one patient will be discussed. Lets assume that the patient 'X', who was mentioned in Chapter 2, is a male child, born in 2012, and classified with a red tag and he has a broken arm. His initial diagnosis will be Orthopedic. One of ambulance personnel enters his information into the search interface. After clicking the *Find a hospital* button, the information is saved on the Patients table in the database server. Then, the search process begins by accessing the XML file. By calculating the patient's age, the XPath query is defined to search for any hospital that has an ”Any” or ”Red” value in its *tag* node, ”Any” or number less than patient's age in its *age* node, and ”Any” or ”Orthopedic” in the *diag* node. From the policy file that is shown in Figure 3.3, since only four hospitals meet the patient's status criteria, the result of the XPath query will return only four hospital IDs: 2,3,4, and 6. The values of the ID attributes are used in the WHERE statement to build an SQL query. As a result, the complex evaluation goes

```
<BasicAttributes>

  <Hospital id="1" name="Strong Memorial">
    <tag>any</tag>
    <age>any</age>
   <diag>Cardiac</diag>
   <diag>Seizures</diag>
  </Hospital>

  <Hospital id="2" name="Unity Ob/Gyn">
    <tag>Red</tag>
    <age>10</age>
    <diag>any</diag>
  </Hospital>

  <Hospital id="3" name="Rochester Immediate Care">
   <tag>any</tag>
    <age>any</age>
   <diag>Orthopaedic</diag>
  </Hospital>

  <Hospital id="4" name="Highland">
    <tag>any</tag>
    <age>any</age>
    <diag>any</diag>
  </Hospital>

  <Hospital id="5" name="Park Ridge">
    <tag>Yellow</tag>
    <tag>Green</tag>
    <age>10</age>
   <diag>Seizures</diag>
  </Hospital>

  <Hospital id="6" name="Rochester General">
    <tag>any</tag>
    <age>any</age>
    <diag>Cardiac</diag>
   <diag>Orthopaedic</diag>
  </Hospital>

  <Hospital id="7" name="Unity Hospital">
    <tag>Yellow</tag>
    <tag>Green</tag>
    <age>any</age>
    <diag>any</diag>
  </Hospital>
</BasicAttributes>
```

Figure 3.3: XML file.

through four hospitals information instead of seven.

When fetching the information of the chosen hospitals from Hospitals table, the system calculates the distance between the location of Patient 'X' and the location of each hospital using their latitude and longitude. The process of calculating distance is done on the database server. After sending the latitude and longitude of both Patient 'X' and each hospital, the function returns the distance in kilometers. As a result, the web server is able to receive the hospitals list, which is ordered by the nearest hospital name to the patient's location. At this point, the web server has access only to hospitals' information that their policy has been passed the first step in the evaluation. The fetched information is used for the second and final evaluation to choose the most appropriate hospital for Patient 'X'. Logically, the nearest hospital is the first choice for any patient, especially one who is classified with a red tag. Therefore, the second evaluation begins for the nearest hospital to the patient's location. The time and road distance is calculated by assuming the speed of the ambulance at 45 mph. The policy of the second evaluation that is done in this implementation is as follows: (Figure 3.4 details the second-step evaluation policy.)

- If the patient is classified as a red tag, then the time of road distance should be less than the time that was defined as an approximate holding time in the ED for the patient.

- If the patient is classified as a red tag, then the number of waiting patients who are classified as red tags should be less than the number of bed capacity in the ED that was defined in the hospital's information.

- If the patient is classified as a yellow tag, and the time of road distance is more than the time that was defined as an approximate holding time for the patient, then the number of waiting patients should be less than the number of bed capacity in the ED that was defined in the hospital's information.

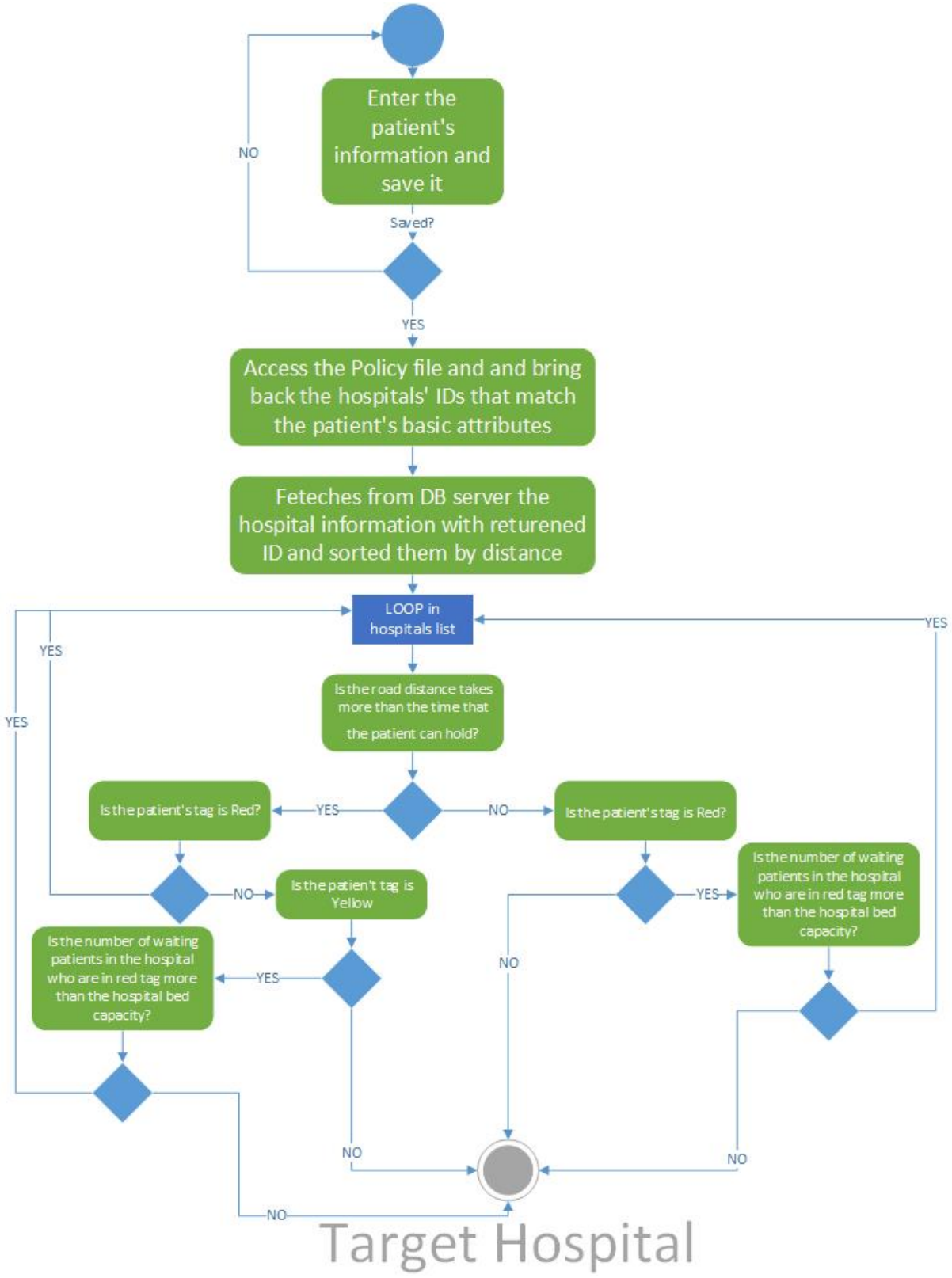

Figure 3.4: The complete chart of system policy.

3.3 Database Server

The database server is the backend of the application. The application uses a remote database server. The server type and version are MySQL 5.1. All data in the database server are stored in tabular format. All of the hospitals' information, except their policies, are stored on the database server on HOSPITALS table. In addition, all patients information are saved on the PATIENTS table. Figure 3.5 shows the database structure.

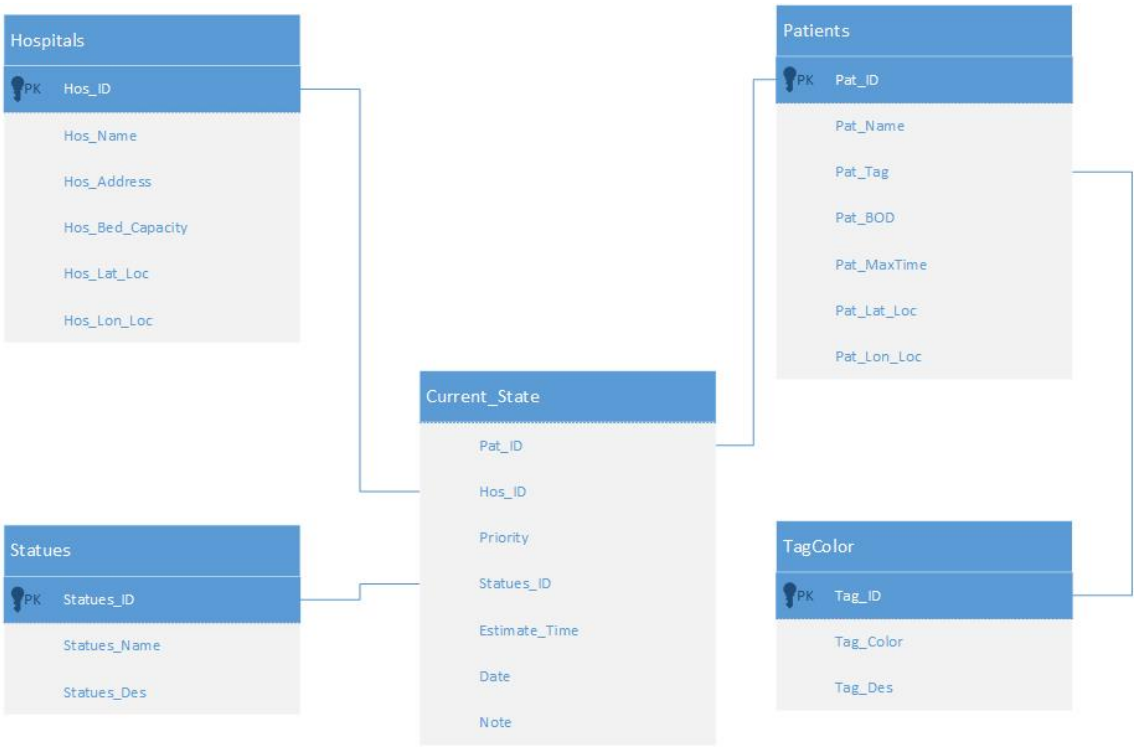

Figure 3.5: The database structure.

Besides hospital and patient information, all patient movements during the crisis time are stored on the database server. For each patient, the patient's information is saved on the PATIENTS table, then, after finding the most appropriate hospital for that patient,

the patient's and hospital's ID are saved on the CURRENT_STATUS table. The CURRENT_STATUS table provides the waiting list of each hospital including the patient's priority, status, and the estimated time for the patient's treatment. The priority is a numeric field that sorts the patients based on their tag color first, then on their arrival time. For example, if there are three patients on the waiting list of one hospital and they are classified as a Red, Yellow, and Green tag. Regardless of their arrival sequence, the patients' priority in being attended to will be 1, 2, and 3, respectively. When a new patient is put on the waiting list, the priority number will be different depending on his/her tag color. So, the priority of the new patient will be 2 if he/she classified as a Red tag, and the other two patients' priority will shifted to 3 and 4. However, the priority of the new patient will be 3 if he/she is classified as a Yellow tag, and the priority of the patient with the Green tag will be shifted to 4, and so on. The status field demonstrates whether the patient is still on the waiting list or if he/she has already received treatment. The values of the status field are WAITING or DONE. Thus, the number of patients who are on the waiting list will be precisely counted each time the hospital information is called for in the second-step evaluation. Although they are taken into consideration, the priority and status fields are not completed in the implementation because of lack of time.

In addition to the stored data, the database server is responsible for calculating the distance between each hospital and the patient's location. When patient information is entered, a list of initial hospitals' IDs are selected by the XML policy file. Then, this list is used on the WHERE clauses to fetch the hospitals information from the database server. In the SQL query, the distance function calculates the distance in kilometers by using the latitude and longitude values for both the patient's location and each hospital's location. The hospitals are sorted by the distance value.

Chapter 4

Analysis

4.1 Approach

Given that the thesis's hypothesis concerns the ability of the BLAC mechanism to provide a flexible approach to handling situational awareness in a critical infrastructure, the proposed model was examined the handling of a situational awareness in a multi-casualty incident by using empirical data. The model's analysis aimed to demonstrate the ability of a two-step evaluation to provide flexibility in the decision-making of a patients' transportation.

The experiment was designed to measure and compare four different factors in two different approaches. During the experiment, each process goes through the overall policy twice, once after using XML policy file, and then, again, without using the XML policy file. The results in each process were captured and saved on the CSV file. The purpose of this step was to make a comparison between the four factors: Time, Data Usage, Accuracy, and Flexibility. To measure the four factors, they had to be defined based on the thesis criteria. The time measurement function measures the average time that the system takes to pick a hospital for each patient. The data usage measures the ratio of volume of data that is fetched from database and used on the web server with the total data that is saved on the database server. Accuracy, in this experiment, is concerned with the matching between the patient's attributes and the selected hospital. Naturally, when the system uses the XML file, the accuracy is always perfect because the system will only process the data that matches the patient's attributes. Therefore, accuracy, here, measures the percentage of correct decisions that are made without using the XML policy file. Finally, flexibility is concerned with the

ability to change the criteria of the decision making during the time of crisis.

Initially, the information for seven hospitals was entered using the hospital interface (Figure 4.1). The first part of the hospital information was saved on the database server while the second part was saved on the XML policy file on the web server. Table 4.1 summarizes the information that was inserted on hospitals including the values on the database server and the values on the XML policy file. The values of Tag color, Age, and Initial Diagnosis, which are shown in Table 4.1 represent the initial values for each hospital. These values are not fixed and the can be changed during the process, as shown in the various experiments.

Table 4.1: Hospitals' Bed Capacity and Policy

The DB Server		The XML Policy File		
Hospital Number	Bed Capacity	Tag Color	Age	Initial Diagnosis
1	8	Any	Any	Cardiac, Seizures
2	7	Any	10	Any
3	7	Any	Any	Orthopedic
4	8	Any	Any	Any
5	6	Yellow, Green	10	Seizures
6	7	Any	Any	Orthopedic
7	6	Yellow, Green	Any	Any

The experiment was done by using a generic script [20] that had the ability to read any web page configuration. Javascript was used to make up a scenario of entering information of certain number of patients who were affected by a multi-casualty incident. The script generated random information for each patient. The *Math.random()*method was used to generate random tag color and initial diagnosis value, while a custom function was used to generate a random date for the patient's date of birth. All of the patients had the same location latitude and longitude values for each experiment because it was assumed that this was a single incident. The distance between the patients' locations and each hospital's location was calculated using the latitude and longitude of the patient. After completing all fields in the patients' interfaces (Figure 4.2), the script used the *Click* command to click

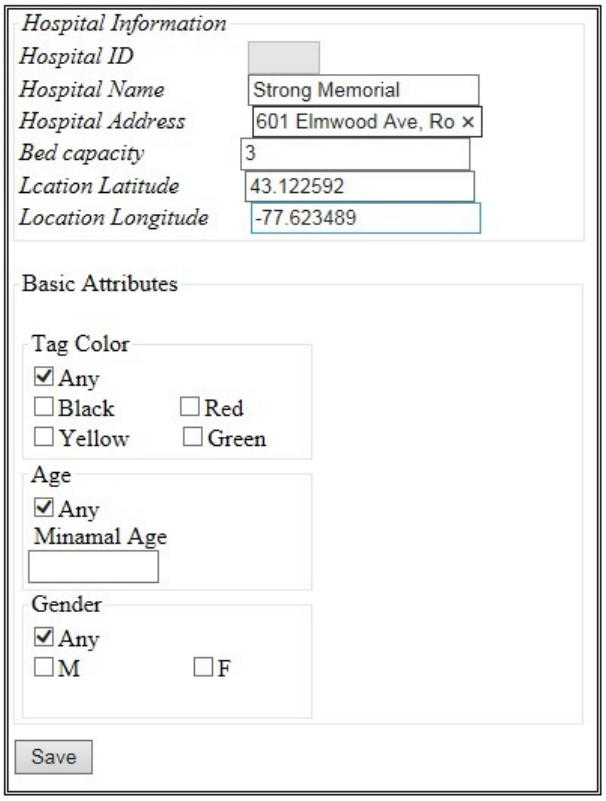

Figure 4.1: Hospitals' interface.

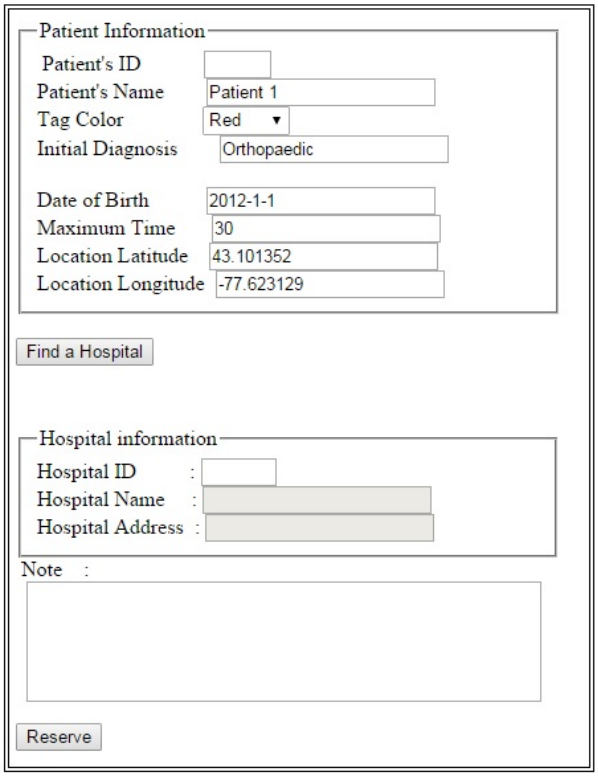

Figure 4.2: Patients' interface.

the *Find a Hospital* button. Then, the *Wait* command displayed until the page finished loading the hospitals' information before the program clicks the *Reserve* button that saves both patients' and hospitals' information on the CURRENT_STATUS table.

The experiment was repeated four times. Each experiment had a different number of patients with random information, and different locations for the hospitals. Moreover,In addition, in each experiment had a different scenario based on the status of the hospitals during the time of the crisis time.

4.2 Analysis

Since time is considered crucial in any MCI, the decision making should be balanced between speed and accuracy, and that can be achieved by reviewing only the relevant data. On this basis, the analysis was concerned about the four factors that were defined previously in Section 4.1, which are: Time, Data Usage, Accuracy, and Flexibility. Although the experiment was repeated and analyzed four times, the analysis of the first experiment was detailed completely in Section 4.2.1, while the remainder only recorded the analyzed results in Table 4.3. Section 4.2.2 analyzed the results of all of the experiments.

4.2.1 Analysis of a Single Experiment

The initial values of the hospital are shown in Table 4.1. The number of patients in this experiment was 80, and the location of incident arranged the hospitals in ascending order based on their distance to the incident, which means that hospital number 1 was the closest hospital to the incident location, while the hospital number 7 was the farthest. The experiment began by changing the policies of the closet two hospitals (1 and 2). The value of their *tag* node was changed to *red* to make them completely occupied for immediate cases. After inserting 60 patients, it was observed that the hospital 1 became overcrowded by having 8 patients who were classified as red tags. As a result, the *tag* node of hospital 1s policy was changed to *NONE* to exclude it from any incoming cases until it was able to dismiss some of its patients. The experiment completed inserting the remaining 20 patients.

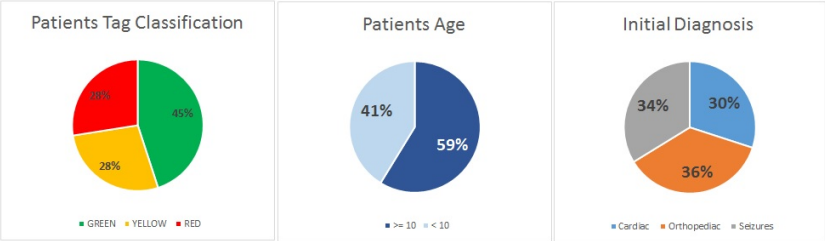

Figure 4.3: Patients data

After completing the process, all of the results were analyzed. As it was pointed out, the patients' information was generated randomly. Figure 4.3 shows the data of patients' information in this experiment. Moreover, the distribution among the hospitals was analyzed in both cases, with and without using the XML policy file. As shown in Figures 4.4, it is clear that hospital numbers 1 and 2 were occupied completely for red-tagged patients. In addition, the hospital numbers 5 and 6 were excluded from any red-tagged patients as they specified in their policy. The distribution in Figure 4.4 could be changed depending on two factors: first, the policy that was made for each hospital (first-step evaluation), and second, the overall policy during a time of crisis (second-step evaluation). The hospitals' policies were easy to change when they became overcrowded and could not receive any more patients as was done with hospital 1 in this experiment. On the other hand, Figure 4.5 shows the distribution of patients when the system did not use the XML policy file. In that case, only the overall policy was considered, regardless of the individual policy for each hospital. As a result, the probabilities of making a wrong decision increased. For instance, 17% of the patients with a red tag were directed to be transported to hospital 5, even though it was not capable of handling red-tagged cases. In addition, the nearest hospital to the incident location (hospital 1) became overcrowded by patients who were classified with a green-tag status, where it should have been maintained for urgent, more serious cases.

Besides the patient distribution and flexibility of changing the hospitals' policies, the time, data usage, and accuracy were measured. The elapsed time was measured in both

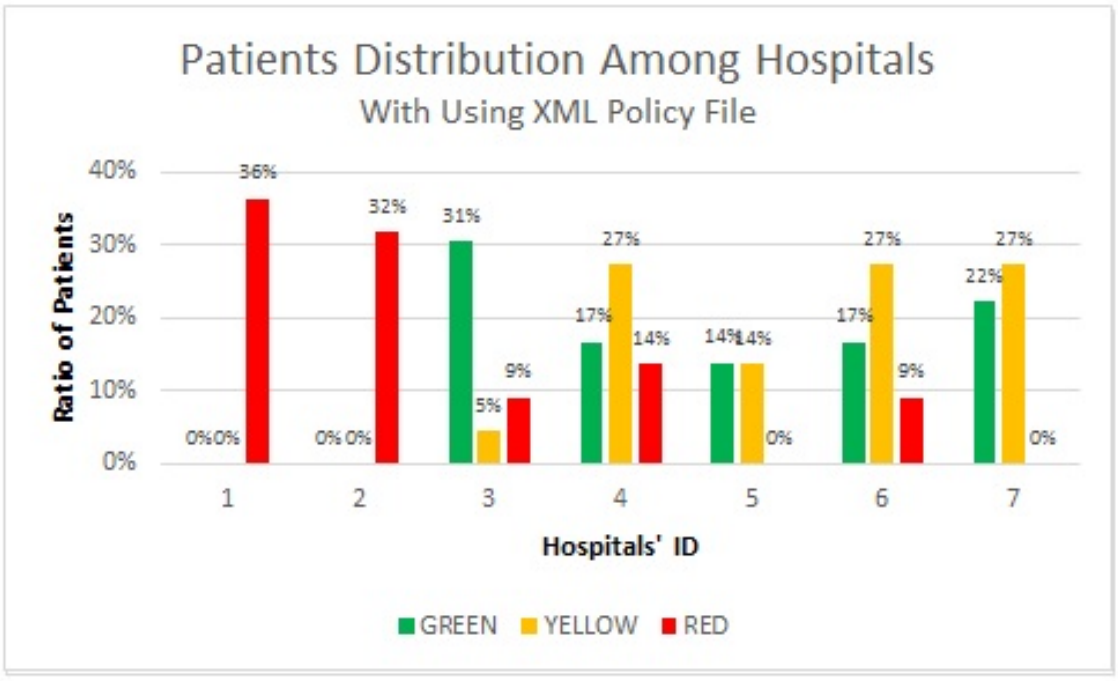

Figure 4.4: Patients' distribution with using XML policy file (Experiment 1).

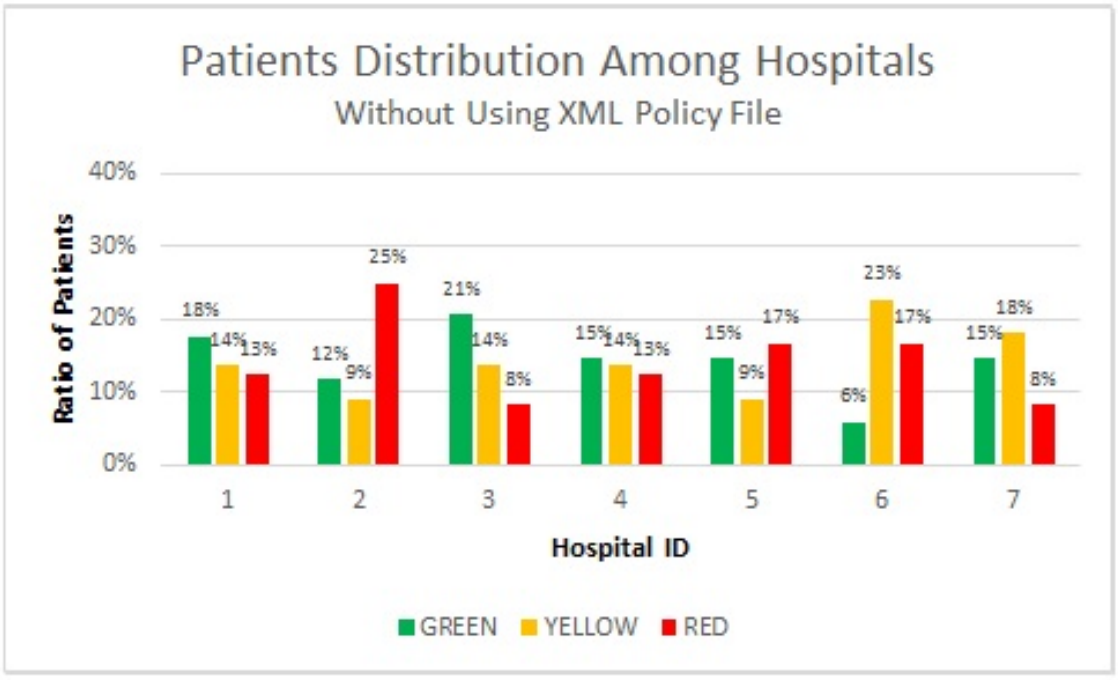

Figure 4.5: Patients' distribution without using XML policy file (Experiment 1).

cases in milliseconds. The average time for the system to find a hospital for a patient was almost the same with and without using the XML policy file. However, a difference emerged in the other two factors, data usage and accuracy. Data usage, as it is defined previously, is concerned with the ratio of data that was fetched and reviewed during the process and the total data in the database server. As shown in Figure 4.6, the search process was done on only 44% of the data, a fact which indicates the ability to impact the time factor on a massive amount of data. In contrast, the system always fetched and reviewed all of the data that was saved on the database server when the XML policy file was not used. On the other hand, there were accuracy concerns about matching the patent's attributes with the chosen hospital. In the case of using the XML policy file, the accuracy always met the definition criteria because the system only fetched and reviewed the data that matched the patient's attributes. On this basis, the decision making was always accurate when the system uses the XML policy file. On the contrary, the decision making could be wrong if the system reviewed all of the data in the initial step. Thus, the accuracy decreased to 48% when the system did not use the XML policy file.

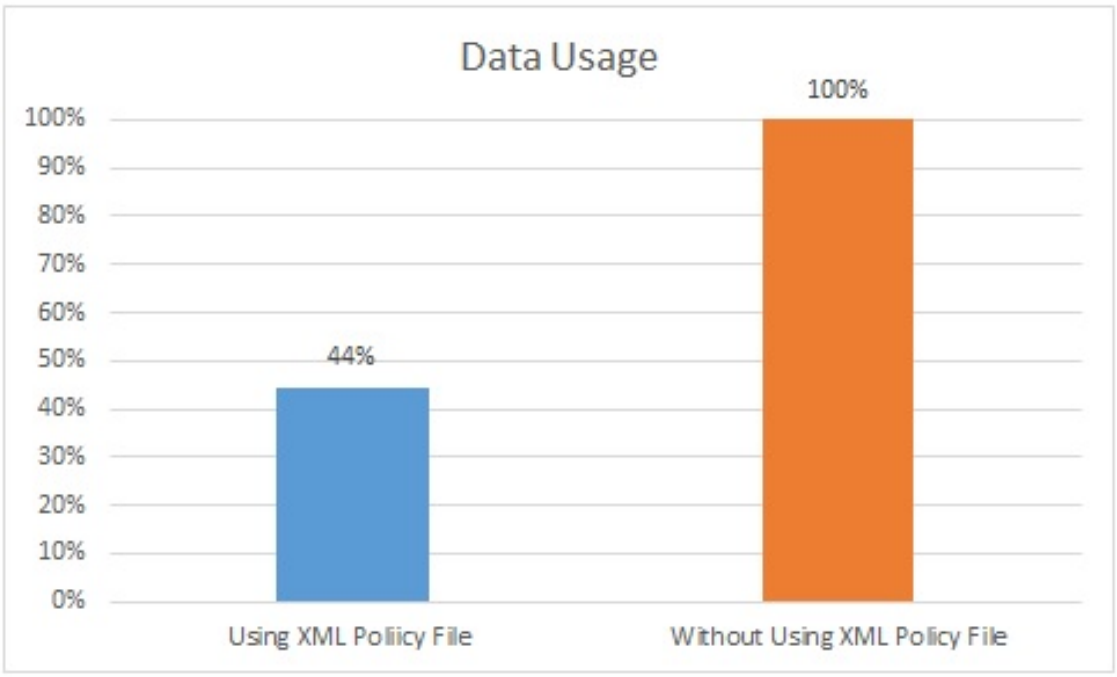

Figure 4.6: Data usage (Experiment 1).

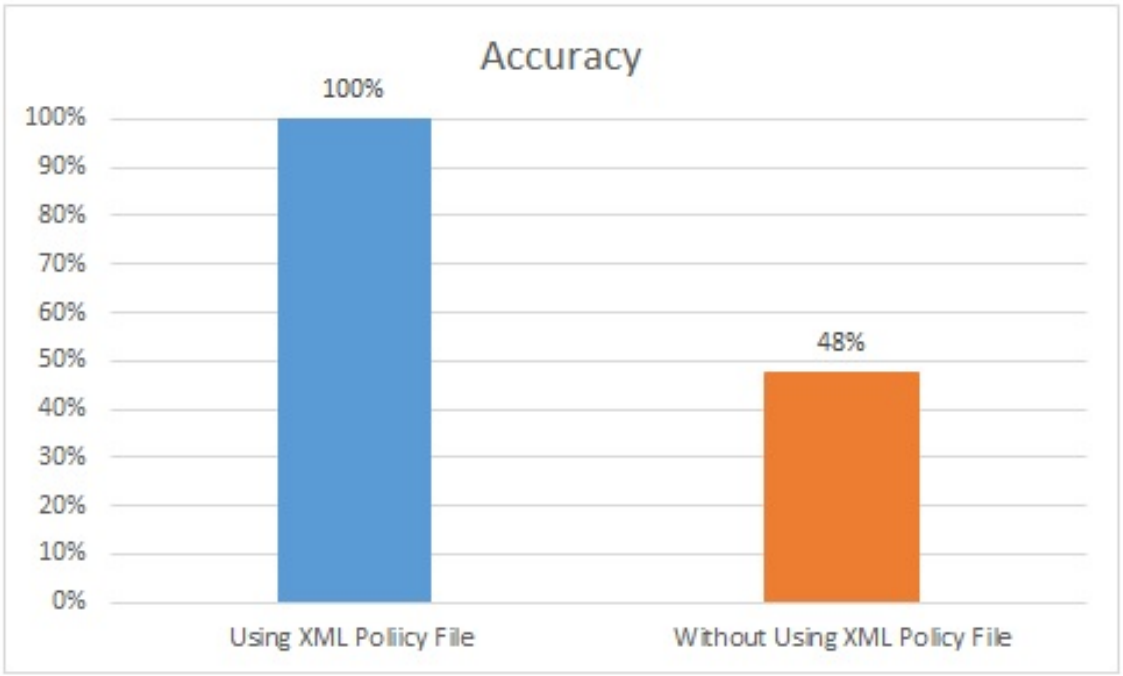

Figure 4.7: Decision-making accuracy (Experiment 1).

4.2.2 Analysis of All Experiments

The experiment was repeated four times. Each experiment had a different number of patients with random information and different locations. Table 4.3 shows the results of the four experiments. The first column has the number of the experiment. Experiment number 1 was explained in detail in Section 4.2.1. Column [*# Pat*] has the number of patients that were used in the experiment. The next three columns ([*Time ms*]: time in milliseconds, [*Data*]: data usage, [*Accuracy*]) have two values: [*W/*], which represents the value when the system used the XML policy file, and [*W/O*] when the system did not use the XML policy file. Column [*Hos*] shows the order of nearest hospitals to the incident location. For example, the nearest hospital to the incident location in experiment 1 was hospital 1, and the farthest hospital was hospital 7. The last column provides a brief description about the scenario that took place during the experiment. The patients' distribution charts for all four experiments were shown in appendix A.

After repeating the experiment four times, the four factors was measured. The flexibility is shown during each experiment when changing the hospital policies depending on the situation. Without using XML policy file, the criteria of accepting any patient would be embedded to the overall policy which is not possible to change during the crisis time. Time, data usage, and accuracy factors were measured after experiment by taking the average of each factor when and without using XML policy file in each experiment.

Table 4.2: T-Test Result Table

	Time		**Data Usage**		**Accuracy**	
	W/	W/O	W/	W/O	W/	W/O
Mean	51.65	51.72	3.16	7	1	0.51
Variance	0.0039	0.0111	0.2030	0	0	0.0118
Observations	4	4	4	4	4	4
Hypothesized Mean Difference	0		0		0	
df	6		6		6	
t Stat	-1.18556780938104		-17.0334893992147		9.02163125540356	
P(T $\leqslant$ t) two-tail	0.280613018		2.62E-06		0.000103856	
t Critical two-tail	2.446911851		2.446911851		2.446911851	

Table 4.3: Experiment Details

#	# Pat	Time ms		Data		Accuracy		Hos	Experiment's Details
		W/	W/O	W/	W/O	W/	W/O		
1	80	51.68	51.66	44%	100%	100%	48%	1 2 3 4 5 6 7	The policy of the nearest two hospitals are modified the value of their tag node to only *RED* to make them completely occupied for immediate cases. After inserting 60 patients, the value of *TAG* node of the Hospital 1 in its policy is changed to *NONE* because the overcrowded on it. Consequently, the system will exclude hospital 1 for the next 20 patients.
2	70	51.70	51.67	43%	100%	100%	43%	2 4 3 1 5 6 7	The policy of the nearest two hospitals are modified the value of their tag node to only *RED* to make them completely occupied for the immediate cases. The system complete inserting all 70 without any changing during the process. At the end of the process, hospital 2 & 4 reached to their maximal capacity.
3	67	51.66	51.66	39%	100%	100%	46%	6 5 4 1 7 2 3	Although the hospital 5 is one of the nearest hospitals to the incident location, it does not change its policy to be occupied for the immediate cases since it is not capable initially to observe patients with red tag. Instead, the hospital number 6 & 4 were occupied for the immediate cases. After inserting 47 patients, it was observed the hospital 4 has available capacity and it assumed there is no more patients with red tag. Consequently, the policy of hospital 4 was changed to make the value of tag node *Yellow* beside *Red*. Thus, hospital 4 was opining for the rest of patients with yellow tag.
4	15	51.56	51.88	54%	100%	100%	66%	1 2 3 4 5 6 7	The location is the same incident's location in experiment 1. In this experiment, it was assumed the worst case when all patients are in RED category. The hospital policies was maintained as it without any changes since it does not make any difference in this case.

4.3 Hypothesis Evaluation

Although it was restricted to MCIs in the healthcare sector, the proposed model was designed to verify the ability of the BLAC mechanism to be used in situational awareness in CIP. The results of proposed model, which are discussed in the analysis section 4.2 are promising. The two-step evaluation that was derived from the BLAC mechanism had a positive impact on the implementation results, even though there was no significant difference in the time measurement. As it was mentioned previously, time was crucial in this situational awareness; however, it was not the only factor considered. The decision making should be balanced between time and accuracy.

The current implementation was designed for MCIs. It used empirical data for hospital and patient information, and it had two different policies: an overall policy and hospital policies. The overall, general policy that should be followed in any crisis, was used for the second-step evaluation. The details of overall policy that was used in implementation is elucidated in Chapter 3. The hospital policies determined the policy for each hospital, individually, and it was used for first-step evaluation. Accordingly, hospitals were associated by their policies (XML policy file), and the patients were associated with their attributes. The experiment included seven hospitals, and the experiment was repeated three times with different number of patients. Time, data usage, and the decision for accuracy were measured and analyzed.

Using the implementation with and without the XML policy file does not show any significant difference in time measurement. The data size could be one reason why it was a quite small. Despite that, the indicators for the other two factors, the usage of data and decision-making accuracy, have a significant difference. Naturally, usage of data and decision-making accuracy play a role, directly or indirectly, in time measurement when the data is massive. For example, if there are thousands of rows in the hospitals' tables, the time will definitely be affected when the search is done on a portion of these data rather than all. Similarly, wrong decisions, in any case, requires more time to transfer patients to the right hospital. Thus, the results of the experiment support the thesis's hypothesis that the

BLAC mechanism has a positive impact on MCIs in the health care sector. Consequently, the thesis concludes that the two-step evaluation process provides a flexible approach to handling SA in CIP.

Chapter 5

Conclusions

5.1 Current Status

The research question were concerned about the ability of a two-step evaluation, derived from Bi-Layer Access Control, to handle situational awareness in critical infrastructures. It provided background of the three related aspects: Critical Infrastructures, Situational Awareness, and Bi-Layer Access Control.

Because the topic of situational awareness in critical infrastructures is a very wide subject, the thesis answered its question through narrowed-down subject to a single situation in a single infrastructure. A multiple-casualty incident is an unexpected incident that overwhelms emergency medical services resources and requires an awareness to distribute the resources effectively among casualties. The thesis proposed a model for an MCI program that decided the most appropriate hospital for each patient depending on the patients' attributes and the hospitals' policies. A complete work-flow process of the proposed model was provided by assuming a certain scenario. The scenario began with saving the hospitals' information and went through using the system during a time of crisis. An example of one patient was explained for clear understanding.

Although a complete process of the proposed model was provided, the implementation covered only the core of the model, which was the search mechanism, including the hospital and patient interfaces. The proposed model used two types of policies: an overall policy that regulated the procedure of patient transportation, and hospital policy that was specific to each hospital. The overall policy was embedded into the system code, while

the hospitals' policies were external in an XML file. The proposed model was tested in an experiment using empirical data that included seven hospitals' information and their policies. The experiment used a script to add any number of patients with random information, and it was repeated four times with a different number of patients and a different incident location. The first time, the experiment was explained in detail, and all of its results were shown. However, the other experiments only recorded their results. There were four factors that were measured in the experiment: time, data usage, accuracy, and flexibility. Each factor was defined precisely, and based on their definition, the comparison was done on the MCI model system with and without using the XML policy file. It was concluded that using the XML policy file in the system did not make a significant difference in time factor. The data usage and decision-making accuracy, which affected the time factor, had a significant difference between the two approaches.

Regardless the data and policy that were used in the implementation, the proposed model can be applied in any CI sectors by declaring the most important elements and defining the policy that should be taken during the crisis time.

5.2 Future Work

Possible future work based on the thesis work could be divided into two sections: future work on the current implementation to complete and modify it, and future work to extend the mechanism of the proposed model in different domains: either in the healthcare sector but other than an MCI, or in a different critical infrastructure area.

5.2.1 Improve the Current Implementation

Since the implementation of the proposed model is limited to verifying the thesis hypothesis, the complete system could be implemented as it was detailed in Chapter 2 in future work. In the current implementation, only two interfaces were designed, which were the hospital interface and the patient interface. The hospital interface was used for inserting the hospitals' information with their policies. As each new record was added into the database

server, a new hospital node was added to the XML policy file in the web server. The patient interface was the main interface that was used in the time of crisis. The application works on inserting patient information and find an appropriate hospital for that patient. To complete a whole model, the system needs to add other interfaces with specific functionality. The current status of the each hospital should be presented in an interface. The current status of the interface should show the waiting list of the hospitals and provide the ability to change the patient status from WAITING to DONE when he/she received treatment. Therefore, this patient would not be counted anymore in the waiting list.

In addition to the interfaces, the structure of the overall policy could be changed. In the current implementation, the overall policy was embedded in the system's code, which means it was not able to be changed afterward. In contrast, the hospital policies were easy to change, even during the time of crisis, because they were in the external XML file. The code read the values of the hospital nodes in the XML file. Therefore, it would be efficient to make the system able to read the overall policy from an external resource.

5.2.2 Extend General Model

The broadness of the situational awareness in the critical infrastructure provides a massive chance to apply the mechanism of the proposed model to any critical infrastructure other than the healthcare sector. On the other hand, the mechanism of the proposed model could be applied to any healthcare's domain other than MCIs. As it was mentioned previously, the proposed model could be extended to other CI sectors by focusing and understanding how the situational awareness is applies in any target sector. Then, the important elements can be associated with policy and attributes that are needed to be changeable during crisis time.

Bibliography

[1] Vincenza Abate, Ludovica Adacher, and Federica Pascucci. Situation awareness in critical infrastructures. *International Journal of Simulation and Process Modelling*, 9(1):92–103, 2014.

[2] Cristina Alcaraz and Javier Lopez. Wide-area situational awareness for critical infrastructure protection. *Computer*, 46(4):30–37, 2013.

[3] Suhair Alshehri. *Toward Effective Access Control Using Attributes and Pseudoroles*. PhD thesis, Rochester Institute of Technology, 2014.

[4] Microsoft API and reference catalog. *System.Web Namespaces*, 2010 (accessed March 1, 2015).

[5] Safa Attia, Abdelhak Boubetra, and Manel Saad Saoud. Decision making issues related to critical infrastructures interdependencies management. *Journal of Advances in Computer Networks*, 2(1), 2014.

[6] Liliya I Besaleva and Alfred C Weaver. Crowdhelp: application for improved emergency response through crowdsourced information. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1437–1446. ACM, 2013.

[7] Brian Bush, LR Dauelsberg, RJ LeClaire, DR Powell, SM Deland, and ME Samsa. Critical infrastructure protection decision support system (cip/dss) project overview. In *Proceedings of the 23rd international conference of the system dynamics society*, pages 17–21, 2005.

[8] World Wide Web Consortium. *XML Path Language*, 2010 (accessed March 1, 2015).

[9] Carlos Cotta. Effective patient prioritization in mass casualty incidents using hyperheuristics and the pilot method. *OR spectrum*, 33(3):699–720, 2011.

[10] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[11] Mica R Endsley. *Designing for situation awareness: An approach to user-centered design*. CRC Press, 2012.

[12] Jeanne M Fair, RJ LeClaire, ML Wilson, AL Turk, SM DeLand, DR Powell, PC Klare, M Ewers, L Dauelsberg, and D Izraelevitz. An integrated simulation of pandemic influenza evolution, mitigation and infrastructure response. In *Technologies for Homeland Security, 2007 IEEE Conference on*, pages 240–245. IEEE, 2007.

[13] Alan Garner, Anna Lee, Ken Harrison, and Carl H Schultz. Comparative analysis of multiple-casualty incident triage algorithms. *Annals of emergency medicine*, 38(5):541–548, 2001.

[14] Nathan R Hoot and Dominik Aronsky. Systematic review of emergency department crowding: causes, effects, and solutions. *Annals of emergency medicine*, 52(2):126–136, 2008.

[15] Shaofeng Liu, Alex HB Duffy, Robert Ian Whitfield, and Iain M Boyle. Integration of decision support systems to improve decision support performance. *Knowledge and Information Systems*, 22(3):261–286, 2010.

[16] Javier Lopez, Roberto Setola, and Stephen D Wolthusen. Overview of critical information infrastructure protection. In *Critical Infrastructure Protection*, pages 1–14. Springer, 2012.

[17] Ramon Martí, Sergi Robles, Abraham Martín-Campillo, and J Cucurull. Providing early resource allocation during emergencies: The mobile triage tag. *Journal of Network and Computer Applications*, 32(6):1167–1182, 2009.

[18] Hyeung-Sik J. Min, Walter Beyeler, Theresa Brown, Young Jun Son, and Albert T. Jones. Toward modeling and simulation of critical national infrastructure interdependencies. *IIE Transactions*, 39(1):57 – 71, 2007.

[19] John D Moteff. *Critical infrastructures: Background, policy, and implementation*. DIANE Publishing, 2010.

[20] David Nevin. *David Nevin Information Systems*, 2015 (accessed April 1, 2015).

[21] Neha Padmanabhan, Frada Burstein, Leonid Churilov, Jeff Wassertheil, Bernard Hornblower, and Nyree Parker. A mobile emergency triage decision support system evaluation. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 5, pages 96b–96b. IEEE, 2006.

[22] Dennis R Powell, Sharon M DeLand, and Michael E Samsa. Critical infrastructure protection decision making. *Wiley Handbook of Science and Technology for Homeland Security*, 2008.

[23] M Samsa, J Van Kuiken, M Jusko, et al. Critical infrastructure protection decision support system decision model: overview and quick-start user's guide. Technical report, Argonne National Laboratory (ANL), 2008.

[24] Melinda Stanners and Han T French. An empirical study of the relationship between situation awareness and decision making. Technical report, DTIC Document, 2005.

[25] George P Tadda and John S Salerno. Overview of cyber situation awareness. In *Cyber Situational Awareness*, pages 15–35. Springer, 2010.

Appendix A

Experiments Charts

The charts of the patients distribution of the four experiments that were made in this thesis are provided in this appendix.

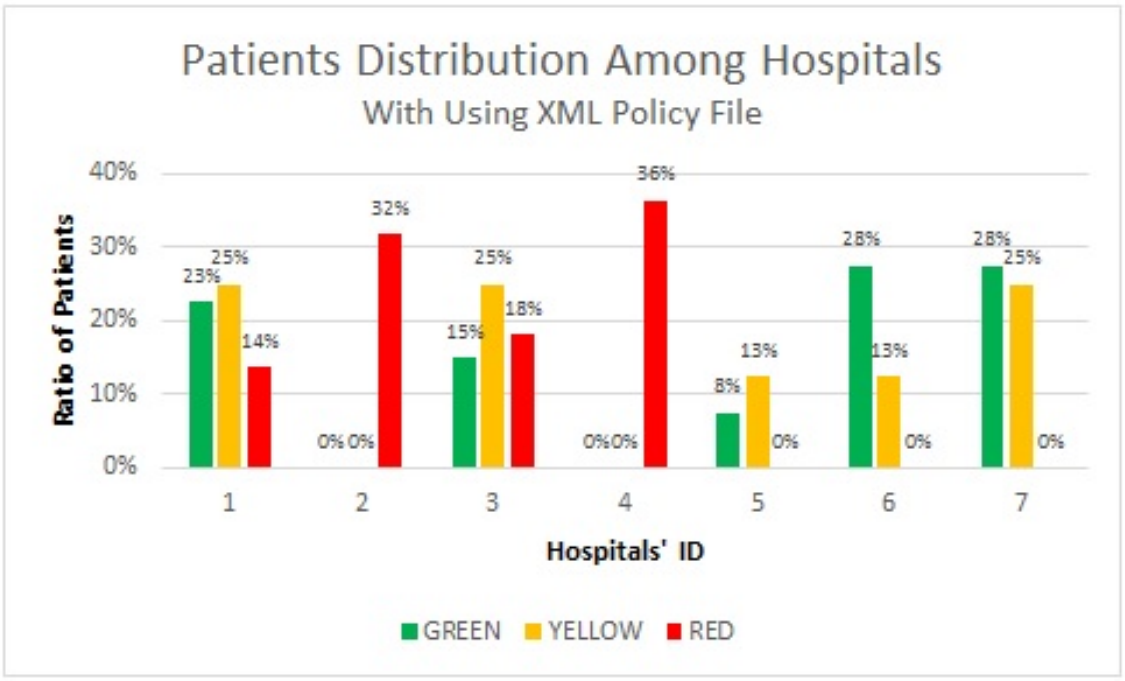

Figure A.1: Patients' Distribution With Using XML Policy File (Experiment 2).

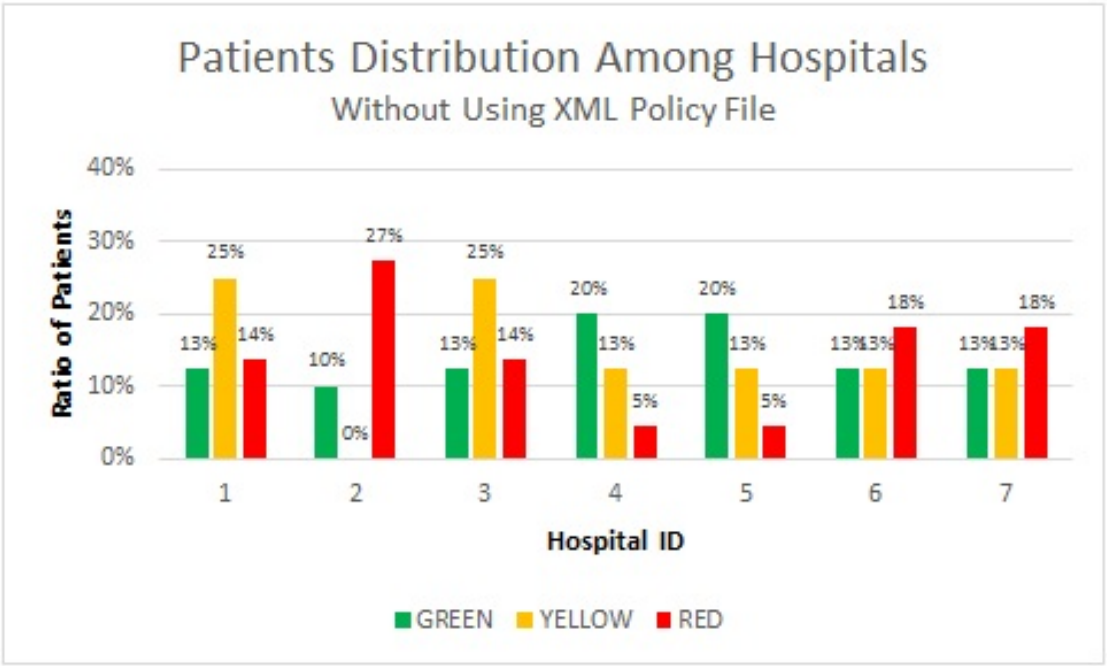

Figure A.2: Patients' Distribution Without Using XML Policy File (Experiment 2).

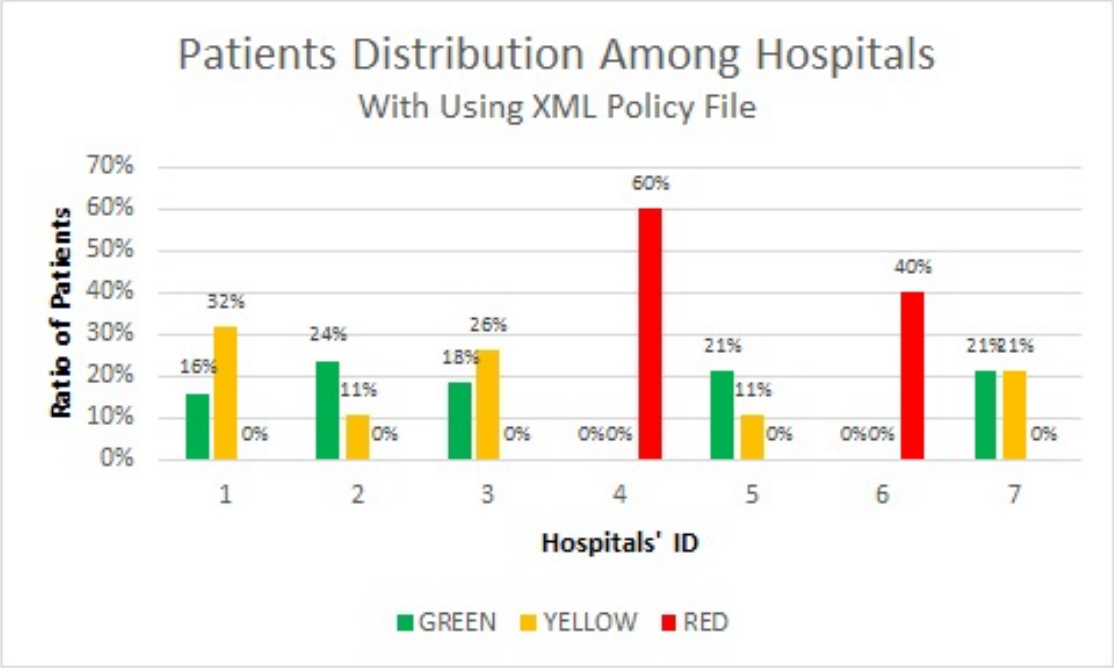

Figure A.3: Patients' Distribution With Using XML Policy File (Experiment 3).

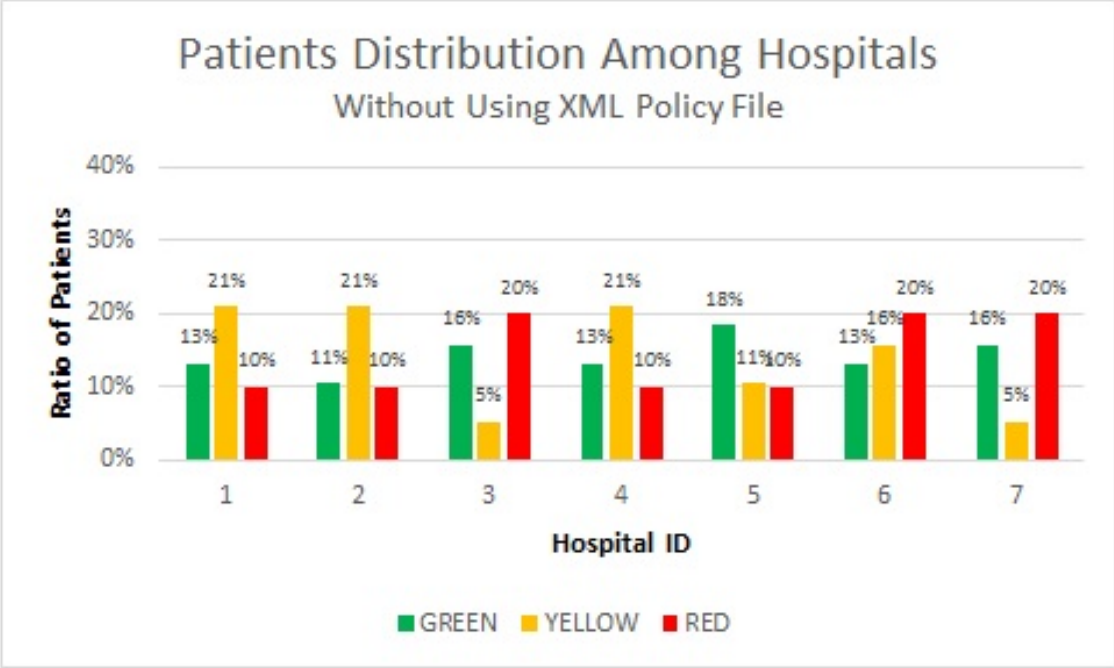

Figure A.4: Patients' Distribution Without Using XML Policy File (Experiment 3).

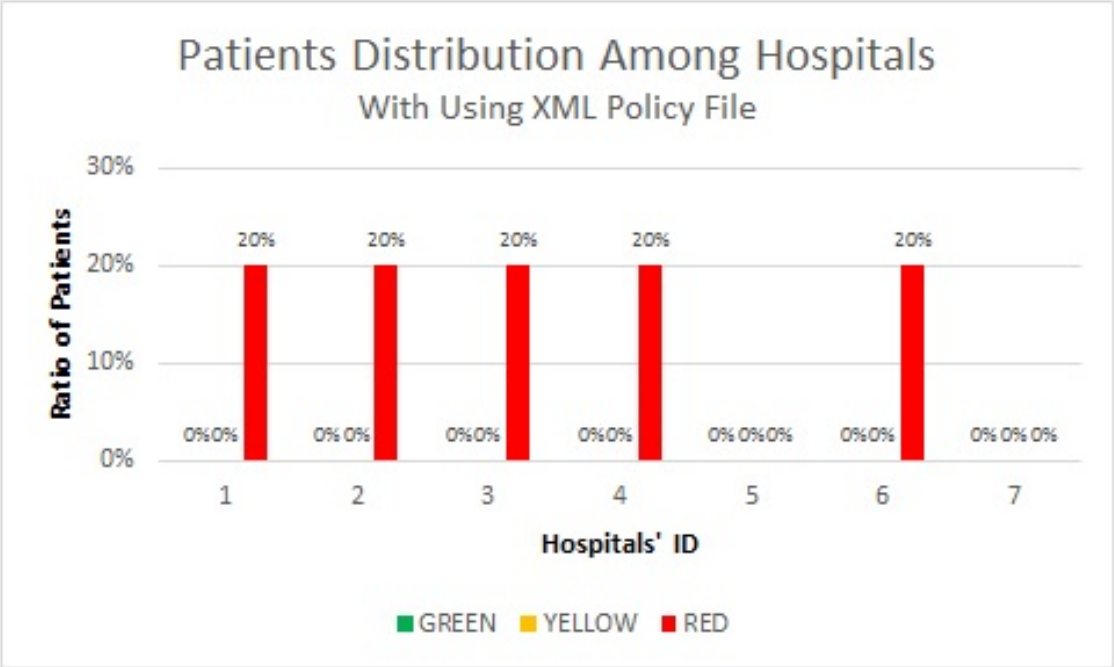

Figure A.5: Patients' Distribution With Using XML Policy File (Experiment 4).

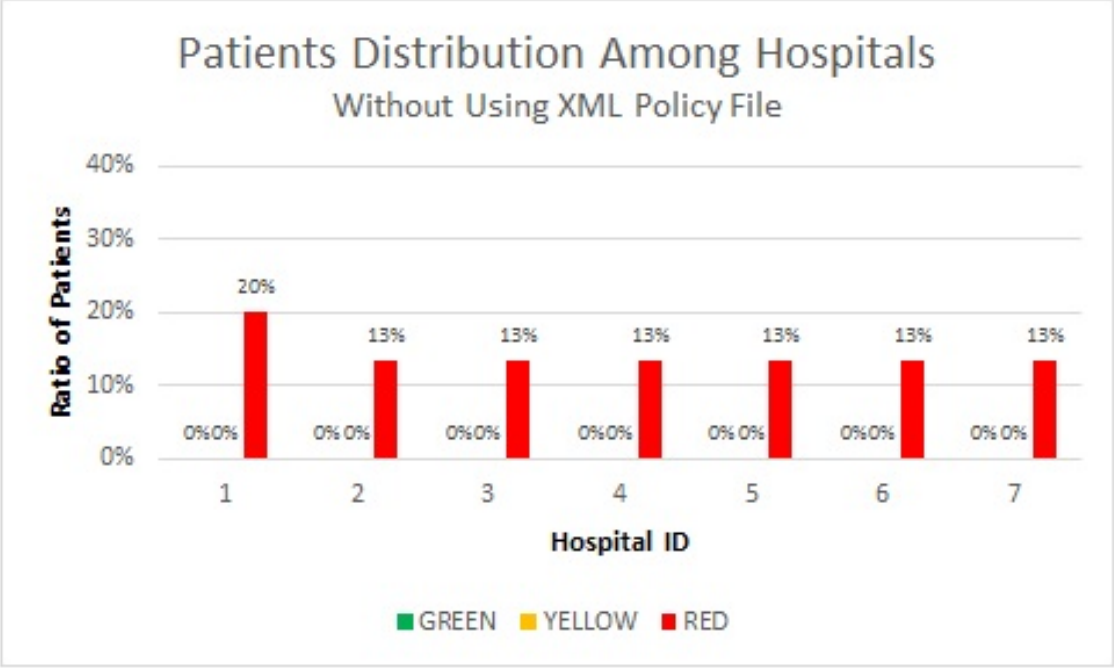

Figure A.6: Patients' Distribution Without Using XML Policy File (Experiment 4).